Final Report

Title of Project: **Quality of Service for Real-Time Applications over Next Generation Data Networks**

PI: Prof. Mohammed Atiquzzaman[1]
Dept. of Electrical & Computer Eng.
University of Dayton, Dayton, OH 45469-0226
Email: atiq@ieee.org

Co-PI Prof. Raj Jain[2]
Dept. of Computer & Information Science
The Ohio State University
Columbus, OH 43210-1277
Email: Jain@CIS.Ohio-State.Edu

Organization: University of Dayton
Ohio State University

Date of Submission: January 15, 2001

Submitted to: NASA Glenn Research Center,
Attn: Dr. William Ivancic
Satellite Networks & Architectures Branch,
M/S 54-2,
Tel: 216 433 3494

Program of Funding: Cooperative Research Agreement

Grant Number: NAG 3-2318

[1] Currently with School of Computer Science, University of Oklahoma, Norman, OK 73019-6151, USA. Tel: (405) 325 8077, Fax: (405) 325 4044, Email: atiq@ou.edu

[2] Currently with Nayna Networks, 157 Topaz St., Milpitas, CA 95035, USA. Email: raj@nayna.com

Introduction

This project, which started on January 1, 2000, was funded by NASA Glenn Research Center for duration of one year. The deliverables of the project included the following tasks:

- Study of *QoS mapping* between the edge and core networks envisioned in the Next Generation networks will provide us with the QoS guarantees that can be obtained from next generation networks.

- *Buffer management* techniques to provide strict guarantees to real-time end-to-end applications through preferential treatment to packets belonging to real-time applications. In particular, use of ECN to help reduce the loss on high bandwidth-delay product satellite networks needs to be studied.

- Effect of *Prioritized Packet Discard* to increase goodput of the network and reduce the buffering requirements in the ATM switches.

- Provision of new *IP circuit emulation services over Satellite IP* backbones using MPLS will be studied.

- Determine the architecture and requirements for *internetworking ATN and the Next Generation Internet* for real-time applications.

Progress

The work of this project has been reported in the following six papers/reports, as listed below. Copies of all the papers are attached to this final report.

1. H. Su and M. Atiquzzaman, "End-to-end QoS for Differentiated Services and ATM Internetworking", *9th International Conference on Computer Communication and Network, October 16~18, 2000, Las Vegas, Nevada.*

2. H. Bai, M. Atiquzzaman and W. Ivancic, “Achieving End-to-end QoS in the Next Generation Internet: Integrated Services over Differentiated Service Networks”, submitted to *2001 IEEE Workshop on High Performance Switching and Routing*, May 29-31, 2001, Dallas, Texas USA.

3. H. Bai, M. Atiquzzaman and W. Ivancic, “Achieving QoS for Aeronautical Telecommunication Networks over Differentiated Services”, submitted for publication as *Technical Report,* NASA Glenn Research Center.

4. A. Durresi, S. Kota, M. Goyal, R. Jain, V. Bharani, “Achieving QoS for TCP traffic in Satellite Networks with Differentiated Services”, Accepted in *Journal of Space Communications*.

5. C. Liu and R. Jain, “Improving Explicit Congestion Notification with the Mark-Front Strategy”, *Computer Networks,* vol 35, no 2-3, pp 285-201, January 2001

6. C. Liu and R. Jain, “Delivering Faster Congestion Feedback with the Mark-Front Strategy”, *International Conference on Communication Technologies* (ICCT 2000), Beijing, China, August 21-25, 2000.

Presentations

The investigators have presented their progress at two presentations at NASA Glenn Research Center. Copies of the slides from the presentation are attached to this final report.

Conclusion

The project has completed on time. All the objectives and deliverables of the project have been completed. Research results obtained from this project have been published in a number of papers in journals, conferences and technical reports.

Acknowledgement

The investigators thank NASA Glenn Research Center for support of this project and Dr. William Ivancic for his guidance and many technical discussions during the course of this project. They also thank Dr. Arian Durressi for his help with the project when the Co-PI was on leave.

Mohammed Atiquzzaman
Principal Investigator and Final Report Editor
School of Computer Science
University of Oklahoma
Norman, OK 73019-6151.
Tel: (405) 325 8077, Fax: (405) 325 4044,
Email: atiq@ou.edu, atiq@ieee.org
Web: http://www.cs.ou.edu/~atiq

Paper 1

H. Su and M. Atiquzzaman, "End-to-end QoS for Differentiated Services and ATM Internetworking"

End-to-end QoS for Differentiated Services and ATM Internetworking[1]

Hongjun Su Mohammed Atiquzzaman
Dept. of Electrical Computer Engineering
University of Dayton, Dayton, OH 45469-0226
Email: suhongju@flyernet.udayton.edu atiq@udayton.edu

Abstract— **The Internet was initially design for non real-time data communications and hence does not provide any Quality of Service (QoS). The next generation Internet will be characterized by high speed and QoS guarantee. The aim of this paper is to develop a prioritized early packet discard (PEPD) scheme for ATM switches to provide service differentiation and QoS guarantee to end applications running over next generation Internet. The proposed PEPD scheme differs from previous schemes by taking into account the priority of packets generated from different application. We develop a Markov chain model for the proposed scheme and verify the model with simulation. Numerical results show that the results from the model and computer simulation are in close agreement. Our PEPD scheme provides service differentiation to the end-to-end applications.**

Keywords— **Differentiated Services, TCP/IP-ATM Internetworking, End-to-end QoS, Queue analysis, analytical model, performance evaluation, Markov chains.**

I. Introduction

With quick emergence of new Internet applications, efforts are underway to provide Quality of Service (QoS) to the Internet. Differentiated Services (DS) is one of the approaches being actively pursued by the Internet Engineering Task Force (IETF) [1], [2], [3], [4], [5]. It is based on service differentiation, and provides aggregate services to the various application classes. DS has defined three service classes. When running DS over ATM (which is implemented by many Internet service providers as their backbones), we need proper services mapping between them. Premium Service requires delay and loss guarantees, and hence it can be mapped to the ATM Constant Bit Rate (CBR) service. Assured Service only requires loss guarantees and hence can be mapped to ATM Unspecified Bit Rate (UBR) service with Cell Loss Priority (CLP) bit set to zero. The Best Effort service does not require any loss or delay guarantee and can be mapped to the ATM UBR service with CLP bit set to one.

It has been shown that Internet may loss packets during high load periods, even worse is that it may suffer congestion collapse [6], [7]. Packets loss means all of the resources they have consumed in transit are wasted. When running DS over ATM, packets loss may lead to more serious results. Because messages will be break into small fix size packet (call cells), one packet loss will lead to the whole message be transmitted again [8]. This makes the congestion scenario even worse. Transmitting useless incomplete packets in a congested network wastes a lot of resource and may result in a very low goodput (good throughput) and poor bandwidth utilization of the network. A number of message based discard strategies have been proposed to solve this problem [8], [9], [10], [11]. These strategies attempt to ensure that the available network capacity is effectively utilized by preserving the integrity of transport level packets during congestion periods. Early Packet Discard (EPD) strategy [8] drops entire messages that are unlikely to be successfully transmitted prior to buffer overflow. It prevents the congested link from transmitting useless packets and reduces the total number of incomplete messages. EPD achieves this by using a threshold in the buffer. Once the queue occupancy in the buffer exceeds this threshold, the network element will only accept packets that belong to a message that has at least one packet in the queue or has already been transmitted. Also per-VC based EPD schemes [12], [13] are proposed to solve the fairness problem that a pure EPD may suffer when virtual circuits compete for the resource. Although EPD can improve the goodput at a network switch, it does not distinguish among priorities of different applications. Previous studies on EPD have assumed a single priority of all ATM psckets, and thus fail to account for the fact that ATM packets could have priority and need to be treated differently. Without a differentiation between the packets, end-to-end QoS guarantee and service differentiation promised by DS networks cannot be ensured when packets traverse through an ATM network. The objective of this study is to developed message based discarding scheme which will account for priority of packets and will be able to provide service differentiation to end applications.

In this paper,we propose a prioritized EPD (PEPD) scheme which can provide the necessary service differentiation needed by the future QoS network. In the PEPD scheme, two thresholds are used to provide service differentiation. We have developed Markov chain models to study the performance of our proposed scheme. The effectiveness of PEPD in providing service differentiation to the two classes of ATM packets coming from a DS network is estimated by the model and then validated by results obtained from our simulation. We measure the goodput, packet loss probability and throughput of the two service classes as a function of the load. Given a QoS requirement for the two service classes, our model can predict the size of the buffer required at the ATM switches and the value of the two thresholds to be used to achieve the target QoS. This model can provide a general framework for analysis of networks carrying messages from applications which require differen-

[1] This work was supported by NASA grant no. NAG3-2318 and Ohio Board of Regents Research Challenge grant

tial treatment in terms of Quality of Service (QoS).

The rest of this paper is organized as follows. Section II lists the assumptions used in the model. Section III constructs a Markov chain model to analyze our proposed PEPD scheme. The model is used to study the performance of the PEPD policy using goodput as the performance criteria. Numerical results from both modeling and computer simulation are presented in Section IV. Concluding remarks are given in Section V.

II. Modeling Assumptions

In the dispersed message model [11], [14], a higher layer protocol data unit (message) consists of a block of consecutive packets that arrive at a network element at different time instants. TCP/IP based systems are examples of such a model. In TCP/IP, the application message is segmented into packets, which are then transmitted over the network. At the receiving end, they are reassembled back into a message by the transport protocol before being delivered to higher layers.

- We assume variable length packets, the length of the packets being geometrically distributed with parameter q (independent between subsequent packets). Clearly, the average packet length is $1/q$ packets. This kind of assumption is typical for data application such as document file and e-mail.
- We also assume that the packets arrive at a network element according to a Poisson process with rate λ, and the transmission time of a packet is exponentially distributed with rate μ. Although we assume that packets are of variable length, Lapid's work [11] shows that this kind of model fits well for fixed-length packet (which is typical to ATM network) scenarios.
- The network element we used in this paper is a simple finite input queue that can contain at most N packets. When the packets arrive at the network element, it enters the input queue only when there is space to hold it; otherwise it is discarded.
- Packets leave the queue according to a first-in-first-out (FIFO) order. When a server is available, the packet at the head of the queue can be served. A packet is transmitted by the server of the network element during its service time. Hence, the network element can be viewed as a M/M/1/N model, with arrival rate λ and service rate μ.
- The input load to the network element is defined as $\rho = \lambda/\mu$.

III. Performance Evaluation of PEPD Scheme

In this section, we describe the PEPD scheme, followed by model setup and performance analysis.

A. PEPD scheme

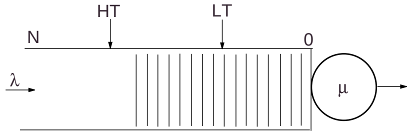


Fig. 1. Network element using PEPD policy.

In the PEPD scheme, we use two thresholds: a low threshold (LT) and a high threshold (HT), with $0 \leq LT \leq N, LT \leq HT \leq N$. As shown in Fig. 1, let QL indicate the current queue length. The following strategy is used to accept packets in the buffer.

- If $QL < LT$, all packets are accepted in the buffer.
- If $LT \leq QL < HT$, new low priority messages will be discarded; only packets belonging to new messages with high priority or packets belonging to messages which have already entered the buffer are accepted.
- If $HT \leq QL < N$, all new messages of both priorities are discarded.
- For $QL \geq N$, packets belonging to all messages are lost because of buffer overflow.

B. Proposed PEPD model

To model the PEPD scheme, we must distinguish between two modes: the *normal mode* in which packets are accepted and the *discarding mode* in which arriving packets are discarded. The state transition diagram for this policy is shown in Fig. 2. In the diagram, state (i, j) indicates that the buffer has i packets and is in j mode, where $0 \leq i \leq N$, $j = 0$ or 1. $j = 0$ corresponds to the normal mode, while $j = 1$ represents the discarding mode. We assume that a head-of-message packet arrives with probability q. The probability that an arriving packet is part of the same message as the previous packets is $p = 1 - q$, and hence is discarded with that probability in the case that that message is being discarded.

According to PEPD, if a message starts to arrive when the buffer contains more than LT packets, the complete new message is discarded if it is of low priority, while if a new message starts to arrive when the buffer contains more than HT packets, the complete message is discarded regardless of its priority. Once a packet is discarded, the buffer enters the discarding mode, and discards all packets belonging to this discarded message. The system will remain in discarding mode until another head-of-message packet arrives. If this head-of-message packet arrives when $QL < LT$, it is accepted, and the system enters the normal mode. If this packet arrives when $LT \leq QL \leq HT$, then the system enters the normal mode only if this packet has high priority. Otherwise, it stays in the discarding mode. Of course, when $QL > HT$, the buffer stays in the discarding mode. Let's assume that h and $l = 1 - h$ be the probabilities of a message being of high and low priority respectively. Also let

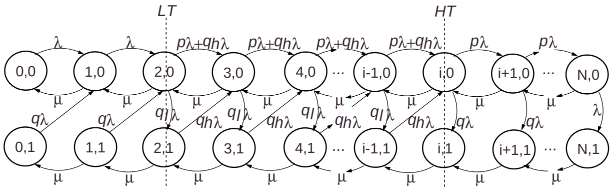


Fig. 2. Steady-state transition diagram the buffers using PEPD

$P_{i,j}(0 \leq i \leq N, j = 0, 1)$ be the steady-state probability of the buffer being in state (i, j). From Fig. 2, we can get the following equations. The solutions of these equations will generate the steady-state probabilities of the buffer states.

$$\begin{aligned}
\lambda P_{0,0} &= \mu P_{1,0} \\
q\lambda P_{0,1} &= \mu P_{1,1} \\
(\lambda + \mu)P_{i,0} &= \lambda P_{i-1,0} + \mu P_{i+1,0} + qh\lambda P_{i-1,1} \\
&\quad\; 1 \leq i \leq LT \\
(\lambda + \mu)P_{i,0} &= (\lambda p + qh\lambda)P_{i-1,0} + \mu P_{i+1,0} \\
&\quad\; + qh\lambda P_{i-1,1} \quad LT < i \leq HT \\
(\lambda + \mu)P_{i,0} &= p\lambda P_{i-1,0} + \mu P_{i+1,0} \quad HT < i < N \\
(\lambda + \mu)P_{N,0} &= p\lambda P_{N-1,0} \\
\mu P_{N,1} &= \lambda P_{N,0} \qquad\qquad (1) \\
\mu P_{i,1} &= q\lambda P_{i,0} + \mu P_{i+1,1} \quad HT \leq i < N \\
(qh\lambda + \mu)P_{i,1} &= q\lambda(1 - h)P_{i,0} + \mu P_{i+1,1} \\
&\quad\; LT \leq i < HT \\
\sum_{i=0}^{N}(P_{i,0} + P_{i,1}) &= 1
\end{aligned}$$

C. Performance analysis of PEPD

In this section, we derive the expression of goodput G for high and low priority messages. The goodput G is the ratio between total good packets exiting the buffer and the total arriving packets at its input. Good packets are those packets that belong to a complete message leaving the buffer. In this paper, we define the goodput for high (or low) priority as the ratio between total number of good packets with high (or low) priority exiting the system and the total number of arriving high (or low) priority packets at the buffer. However, we normalize the goodput to the maximum possible goodput.

Let W be the random variable that represents the length (number of packets) of an arriving message, and V be the random variable that represents the success of a message. $V = 1$ for a good message, and $V = 0$ for an incomplete message. Let U be the random variable that represents the priority of a packet, $U = 1$ for high priority packets and $U = 0$ for low priority packets. The goodput for the high priority packets (G_h) is

$$G_h = \frac{\sum_{n=1}^{\infty} nP(W = n, V = 1, U = 1)}{\sum_{n=1}^{\infty} nP(W = n, U = 1)} \qquad (2)$$

where the numerator represents the total good packets exiting the buffer and the denominator is the total arriving packets at a network input. Note that W and V are independent random variables, and the length of an arriving message is geometrically distributed with parameter q, which means the average length of the messages is $1/q$. Then the denominator of Eq. (2) can be expressed as

$$\sum_{n=1}^{\infty} nP(W = n, V = 1) = P(U = 1)\sum_{n=1}^{\infty} P(W = n) = \frac{h}{q} \qquad (3)$$

Substituting Eq. (3) in Eq. (2),

$$G_h = \frac{q}{h}\sum_{n=1}^{\infty} nP(W = n, V = 1, U = 1) \qquad (4)$$

The probability of an incoming high priority message of length n to be transmitted successfully can be expressed as follows:

$$\begin{aligned}
P(W = n, V = 1, U = 1) &= P(V = 1|W = n, U = 1) \\
&\quad\; P(W = n, U = 1) \\
&= P(V = 1|W = n, U = 1) \\
&\quad\; P(W = n)P(U = 1) \quad (5) \\
&= q(1 - q)^{n-1}h \\
&\quad\; P(V = 1|W = n, U = 1)
\end{aligned}$$

Let Q be the random variable representing the queue occupancy at the arrival of a head-of-message packet. Then

$$\begin{aligned}
P(V = 1|W = n, U = 1) &= \sum_{i=0}^{N} P(V = 1|W = n, U = 1, \\
&\quad\; Q = i)P(Q = i) \qquad (6)
\end{aligned}$$

where $P(Q = i) = P_{i,0} + P_{i,1}$ is the probability of the queue occupancy. $P_{i,j}$ is obtained from from the solution of Eq. (2). By combining Eqs. (4), (5), and (6), we get the goodput of the high priority messages as:

$$\begin{aligned}
G_h &= q\sum_{n=1}^{\infty} nq(1 - q)^{(n-1)}\sum_{i=0}^{N} P(V = 1|W = n, U = 1, \\
&\quad\; Q = i)P(Q = i) \qquad (7)
\end{aligned}$$

Similarly, we can get the goodput for the low priority messages and the total goodput as follows:

$$\begin{aligned}
G_l &= q\sum_{n=1}^{\infty} nq(1 - q)^{(n-1)}\sum_{i=0}^{N} P(V = 1|W = n, U = 0, \\
&\quad\; Q = i)P(Q = i) \qquad (8) \\
G &= q\sum_{n=1}^{\infty} nq(1 - q)^{(n-1)}\sum_{i=0}^{N} P(V = 1|W = n, \\
&\quad\; Q = i)P(Q = i) \qquad (9)
\end{aligned}$$

In order to find the values of G, G_l and G_h, we need to define and evaluate the following conditional probabilities:

$$\begin{aligned} S_{n,i} &= P(V=1|W=n, Q=i) && (10)\\ S_{l,n,i} &= P(V=1|W=n, U=0, Q=i) && (11)\\ S_{h,n,i} &= P(V=1|W=n, U=1, Q=i) && (12) \end{aligned}$$

These conditional probabilities can be computed recursively. Let's take $S_{h,n,i}$ as an example. Consider first a system that employs the PPD policy. Usually, the success of a packet depends on the evolution of the system after the arrival of the head-of-message. However, there is a boundary condition for this. Let us first consider a message of length $1 \leq n \leq N$. Assume that the head-of-message packet belonging to a message of length $n \leq N$ arrives at buffer when $Q = i$. Then, if $i \leq N - n$, there is enough space to hold this message, and this message is guaranteed to be good, i.e

$$\hat{S}_{n,i} = 1 \quad 0 \leq i \leq N-n, \quad 1 \leq n \leq N \qquad (13)$$

note that if $Q = N$ (i.e. the buffer is full), the head-of-message packet is discarded, and the message is guaranteed to be bad. Hence,

$$\hat{S}_{n,N} = 0 \quad 1 \leq n \leq N \qquad (14)$$

Eqs. (13) & (14) give the boundary conditions for this system. For other states of the buffer, we have:

$$\begin{aligned} \hat{S}_{n,i} &= (1-r)\hat{S}_{n-1,i+1} + r\hat{S}_{n,i-1} \\ & \quad N-n+1 \leq i \leq N-1, 1 \leq n \leq N \quad (15) \end{aligned}$$

where $r = \mu/(\mu + h\lambda)$ is the probability that a departure occurs before an arrival. In this case, we only consider the conditional probability for high priority packets, so the arrival rate is $h\lambda$ rather than λ. Eq. (15) can be explained as follows. If the next event following the arrival of a head-of-message packet is the arrival of another packet (which has the probability $1 - r$), this new packet can be viewed as a new head-of-message packet belonging to a message of length $n - 1$. Therefore, the probability that this new message will succeed is $\hat{S}_{n-1,i+1}$. If the event following the arrival of the head-of-message packet is a departure of a packet (which happens with probability r), the probability that the message is successful is $\hat{S}_{n,i-1}$, since it is equivalent to a head-of-message packet that arrived at the system with Q $=i - 1$ packets. So, combining the above two conditions, we can get:

$$\hat{S}_{n,i} = \begin{cases} 1 & N-n+1 \leq i \\ & i \leq N-1 \\ (1-r)\hat{S}_{n-1,i+1} + r\hat{S}_{n,i-1} & N-n+1 \leq i \\ & i \leq N-1 \\ 0 & i = N \end{cases} \qquad (16)$$

For a large message, $n > N$, there is no guarantee that this message will succeed, it's success depended heavily on the evolution of the system after the arrival of the head-of-message packet even for the case of $i = 0$. So, for $n > N$ we get the following equations:

$$\hat{S}_{n,i} = \begin{cases} (1-r)\hat{S}_{n-1,i+1} + r\hat{S}_{n-1,i} & i = 0 \\ (1-r)\hat{S}_{n-1,i+1} + r\hat{S}_{n,i-1} & N-n+1 \leq i \\ & i \leq N-1 \\ 0 & i = N \end{cases} \qquad (17)$$

These recursions are computed in ascending order of both n and i. For a system that employs the PEPD policy, for high priority messages, the above recursions remain correct only when the head-of-message packet arrives at the buffer while the number of packets is below the high threshold, i.e. $Q = i < HT$. For $Q = i \geq HT$, these new messages will be discarded, so

$$S_{h,n,i} = \begin{cases} \hat{S}_{n,i} & i < HT \\ 0 & HT \leq i \leq N \end{cases} \qquad (18)$$

with $r = \mu/(\mu + h\lambda)$. Similarly, we can get

$$S_{l,n,i} = \begin{cases} \hat{S}_{n,i} & i < LT \\ 0 & LT \leq i \leq N \end{cases} \qquad (19)$$

with $r = \mu/(\mu + (1-h)\lambda)$, while the average is

$$S_{n,i} = (1-h)\hat{S}_{l,n,i} + h\hat{S}_{l,n,i} \qquad (20)$$

The above model is used to analyze the performance of PEPD in the next section.

IV. Numerical Results

In this section, we present results from our analytical model and simulation to illustrate the performance of PEPD. We also validate the accuracy of our analytical model by comparison with simulation results. In our experiment, we set $N = 120$ packets, $q = 1/6$ which corresponds to the case where the queue size is 20 times the mean message length. The incoming traffic load (ρ) at the input to the buffer is set in the range of $0.8 - 2.2$, where $\rho < 1$ represents moderate load, and $\rho \geq 1$ corresponds to higher load which results in congestion buildup at the buffer. Goodput of the combined low and high priority packets is defined as $G = h * GH + (1-h) * GL$ as used in Eq. (20).

In order to validate our model, we compare it with results from computer simulation. The simulation setup is simply two nodes compete for a single link with a queue size 120 packets. The two nodes generate messages with a mean length of 6 (measured in packets). Because the queue occupancy is a critical parameter used for calculating the goodput, we compare the queue occupancy obtained from the model and computer simulation in Fig. 3. For $q = 1/6$, it is clear that analytical and simulation results are in close

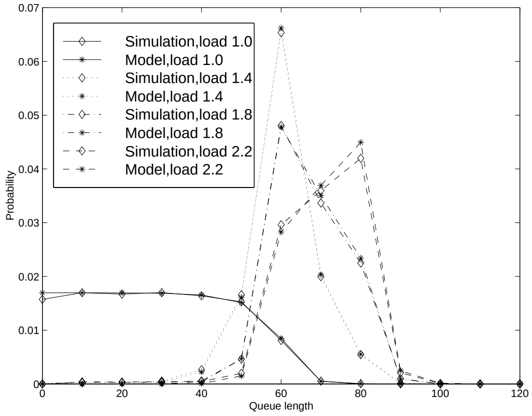


Fig. 3. Comparison of queue occupancy probability from model and simulation with different load. $N = 120, LT = 60, HT = 80, h = 0.5, q = 1/6$.

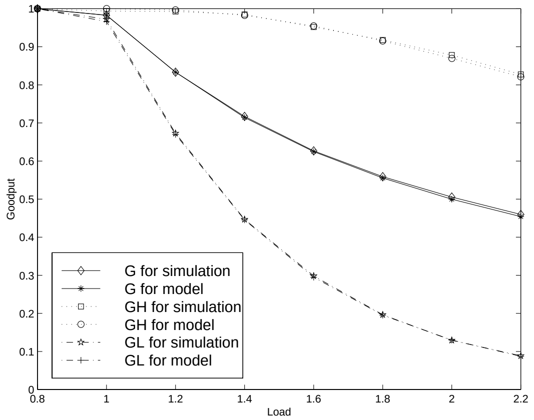


Fig. 4. Goodput versus load for $h = 0.5, N = 120, LT = 60, HT = 80, q = 1/6$. G, GH and GL represents average goodput of all, high priority, and low priority packets respectively.

agreement. Our proposed scheme results in the buffer occupancy varying between LT and HT for even high loads. The exact value depends on the average message length, queue thresholds, etc.

Fig. 4 shows the goodput of the buffer using PEPD for $q = 1/6$ (i.e. mean message length of 6) as a function of the offered load. In this figure, the probability that a message is of high priority is 0.5. From Fig. 4, it is clear that the results from our model and computer simulation fit well. So we conclude that our model can be used to carry out an accurate analysis the PEPD policy. Therefore, in the rest of this section, we will use results from only the model to analyze the performance of PEPD policy.

Fig. 5 shows the goodput for $q = 1/6$ as a function of the offered load and for different mix (h) of high & low priority packets. For a particular load, increasing the fraction of High Priority (HP) packets (h) results in a decrease of throughput of both high and Low Priority (LP) packets. The LP throughput decreases because the increase in h results in fewer LP packets at the input to the buffer in addition to LP packets competing with more HP packets in the buffer space (0 to LT). On the other hand, increase in h results in more HP packets. Since the amount of buffer space (HT-LT) which is reserved for HP packets is the same, the throughput of HP packets decrease. Note that the decrease in the throughput of LP is much faster than the decrease in HP resulting in the overall goodput (as defined by Eq. (20) being constant. Our proposed technique allows higher goodput for high priority packets which may required in scenarios where an application may need a preferential treatment over other applications.

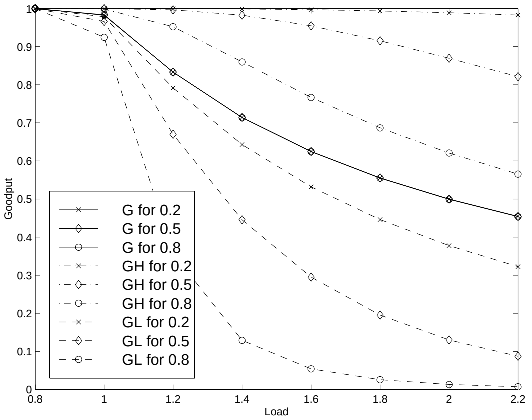


Fig. 5. Goodput versus load for $h = 0.2, 0.5$ and 0.8 with $N = 120, LT = 60, HT = 80, q = 1/6$. G, GH and GL represents average goodput of all, high priority, and low priority packets respectively.

In Fig. 6, we fix LT while varying HT to observe the behavior of the buffer. It is obvious that for a traffic containing fewer high priority packets, increasing the HT will increase the performance of the buffer for high priority packets. This is because increasing HT will let the high priority packets get more benefits from discarding low priority packets, especially for lower values of HT. Increasing HT will result in an initial increase in the goodput for high priority packets followed by a decrease. This is obvious, because for a very high value of HT, the behavior of PEPD will approach that of PPD for high priority packets.

Fig. 7 shows the goodput for high priority message versus the fraction of high priority messages. It is also clear that for a particular load, increasing the high priority traffic will decrease the performance for high priority packets as has been observed in Fig 5.

Finally, in Fig. 8, we keep HT constant while changing LT. For a load of 1.6 and a particular mix of high & low priority packets, we observe that the performance of high priority packets is not very sensitive to a change in LT. However, when LT is set close to HT, the goodput for high priority packets will decrease quickly. This is because when the two thresholds are set too close, the high priority packets do not get enough benefits from discarding low priority packets. We suggest avoiding this mode of operation because the

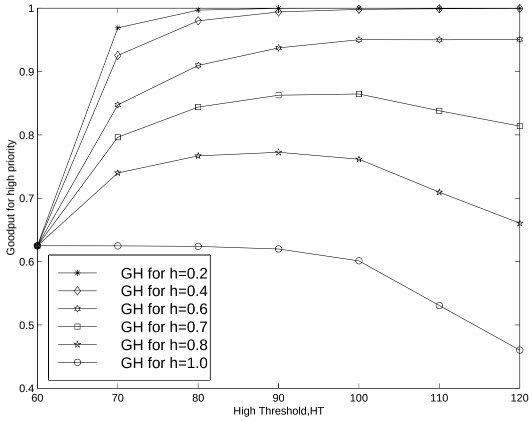


Fig. 6. Goodput for high priority messages versus HT for different h with $LT = 60, q = 1/6$ and input load of 1.6

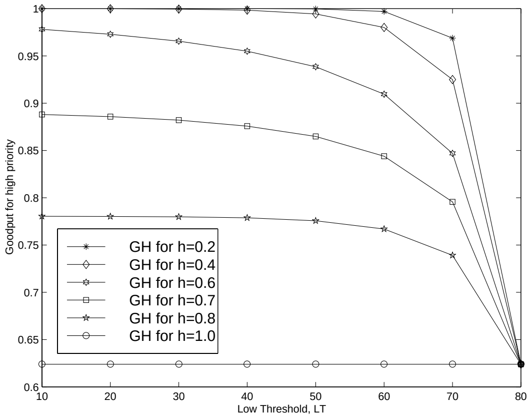


Fig. 8. Goodput for high priority versus LT for $N = 120, HT = 80$, load=$1.6, q = 1/6$.

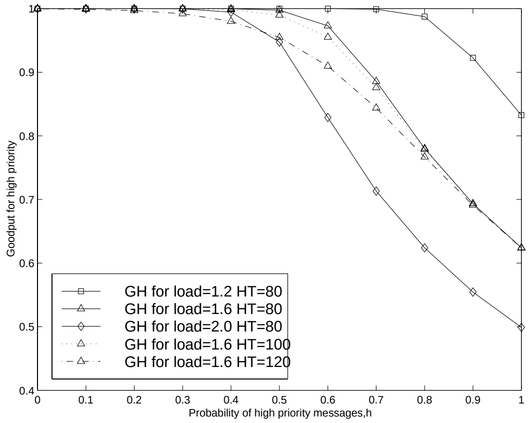


Fig. 7. Goodput of high priority messages versus h for different load and HT for $LT = 60$.

buffer is not fully utilized.

V. Conclusions

In this paper, we have proposed and developed a performance model for the Priority based Early Packet Discard (PEPD) to allow end to end QoS differentiation for applications over Next Generation Internet. To verify the validity of our proposed analytical model, we compared it with results from computer simulation. Numerical results show that the results from the model and computer simulation are in close agreement. The numerical results also show that our proposed PEPD policy can provide differential QoS to low and high priority packets. Such service differentiation is essential to provide QoS to applications running Differentiated service over ATM. Our result show that the performance of PEPD depends on the mix of high & low priority traffic, threshold setting, average message length, etc. Given a certain QoS, the model can be used to dimension the size of the buffer and the PEPD thresholds. Our model can serve as a framework to implement packet based discarding schemes using priority. Results show that this scheme solves some critical problems for running Differentiated Service (DS) over ATM network by ensuring the QoS promised by the Differentiated Service.

References

[1] S. Blake, D. Black, M. Carlson, Z. Wang, and W. Weiss, “An architecture for differentiated services.” RFC 2474, Internet Engineering Task Force, December 1998.

[2] K. Nichols, V. Jacobson, and L. Zhang, “A two-bit differentiated services architecture for the internet.” RFC 2638, Internet Engineering Task Force, July 1999.

[3] K. Nichols, S. Blake, F. Baker, and D.Black, “Definition of the differentiated services field (DS Field) in the IPv4 and IPv6 headers.” RFC 2474, Internet Engineering Task Force, December 1998.

[4] J. Heinanen, F. Blaker, W. Weiss, and J. Wroclawski, “Assured forwording PHB group.” RFC 2597, Internet Engineering Task Force, June 1999.

[5] X. Xiao and L. M. Ni, “Internet QoS: A big picture,” *IEEE Network*, pp. 8–18, April 1999.

[6] J. Nagle, “Congestion control in IP/TCP internetworks.” RFC 896, Internet Engineering Task Force, January 1984.

[7] V. Jacobson, “Congestion avoidance and control,” *Proceedinds of SIGCOMM’88*, pp. 314–329, August 1988.

[8] A. Romanow and S. Floyd, “Dynamics of TCP traffic over ATM network,” *IEEE Journal on Selected Areas in Communications*, vol. 13, pp. 633–641, 1995.

[9] K. Kawahara, K. Kitajima, T. Takine, and Y. Oie, “Packet loss performance of the selective cell discard schemes in ATM switches,” *IEEE Journal on Selected Areas in Communications*, vol. 15, pp. 903–913, 1997.

[10] M. Casoni and J. S. Turner, “On the performance of early packet discard,” *IEEE Journal on Selected Areas in Communications*, vol. 15, pp. 892–902, 1997.

[11] Y. Lapid, R. Rom, and M. Sidi, “Analysis of discarding policies in high-speed networks,” *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 764–772, 1998.

[12] H. Li, K. Y. Siu, H. Y. Tzeng, H. Ikeda, and H. Suzuki, “On TCP performance in ATM networks with per-VC early packet discard mechanisms,” *Computer Communications*, vol. 19, pp. 1065–1076, 1996.

[13] J. S. Turner, “Maintaining high throughput during overload in ATM switches,” *IEEE INFOCOM ’96: Conference on Computer Communications*, San Fransisco, USA, pp. 287–295, Mar. 26-28 1996.

[14] I. Cidon, A. Khamisy, and M. Sidi, “Analysis of packet loss processes in high speed networks,” *IEEE Transactions on Information Theory*, vol. 39, pp. 98–108, 1993.

Paper 2

H. Bai, M. Atiquzzaman and W. Ivancic, “Achieving End-to-end QoS in the Next Generation Internet: Integrated Services over Differentiated Service Networks”

Achieving End-to-end QoS in the Next Generation Internet: Integrated Services over Differentiated Service Networks [1]

Haowei Bai and Mohammed Atiquzzaman [2]

Department of Electrical and Computer Engineering

The University of Dayton, Dayton, Ohio 45469-0226

E-mail: baihaowe@flyernet.udayton.edu, atiq@ieee.org

William Ivancic

NASA Glenn Research Center

21000 Brookpark Rd. MS 54-8,

Cleveland, OH 44135

Abstract

Currently there are two approaches to provide Quality of Service (QoS) in the next generation Internet: An early one is the Integrated Services (IntServ) with the goal of allowing end-to-end QoS to be provided to applications; the other one is the Differentiated Services (DiffServ) architecture providing QoS in the backbone. In this context, a DiffServ network may be viewed as a network element in the total end-to-end path. The *objective* of this paper is to investigate the possibility of providing end-to-end QoS when IntServ runs over DiffServ backbone in the next generation Internet. Our results show that the QoS requirements of IntServ applications can be successfully achieved when IntServ traffic is mapped to the DiffServ domain in next generation Internet.

[1] The work reported in this project was supported by NASA grant no. NAG3-2318

[2] The second author is currently with the School of Computer Science, University of Oklahoma, Norman, OK 73072, Tel: (405) 325 8077, email: atiq@ou.edu

1 Introduction

Quality of Service (QoS) has become the objective of the next generation Internet. QoS is generally implemented by different classes of service contracts for different users. A service class may provide low-delay and low-jitter services for customers who are willing to pay a premium price to run high-quality applications, such as, real-time multimedia. Another service class may provide predictable services for customers who are willing to pay for reliability. Finally, the *best-effort* service provided by current Internet will remain for those customers who need only connectivity.

The Internet Engineering Task Force (IETF) has proposed a few models to meet the demand for QoS. Notable among them are the Integrated Services (IntServ) model [1] and Differentiated Services (DiffServ) [2] model. The IntServ model is characterized by resource reservation. Before data is transmitted, applications must set up paths and reserve resources along the path. The basic target of the evolution of IntServ is to support various applications with different levels of QoS within the TCP/IP (Transport Control Protocol/Internet Protocol) architecture. But IntServ implementation requires RSVP (Resources Reservation Protocol) signaling and resource allocations at every network element along the path. This imposes a bound on its incorporation for the entire Internet backbone.

The DiffServ model is currently being standardized to overcome the above scalability issue, and to accommodate the various service guarantees required for time critical applications. The DiffServ model utilizes six bits in the TOS (Type of Service) field of the IP header to mark a packet for being eligible for a particular forwarding behavior. The model does not require significant changes to the existing infrastructure, and does not need many additional protocols. Therefore, with the implementation of IntServ for small WAN networks and DiffServ for the Internet backbone, the present TCP/IP traffic can meet the present day demands of real time and other quality required traffic. Combining IntServ and DiffServ has been proposed by IETF in [3] [4] as one of the possible solutions to overcome the scalability problem.

To combine the advantages of DiffServ (good scalability in the backbone) and IntServ (per flow QoS guarantee), a mapping from IntServ traffic flows to DiffServ classes has to be performed. Some preliminary work has been carried out in this area. Authors in [5] present a concept for the integration of both IntServ and DiffServ, and describe a prototype implementation using commercial routers. However, they don't present any numerical results. Authors in [6] present results to determine performance differences between IntServ and DiffServ, as well as some characteristics about their combined use.

The *objective* of this paper is to investigate the end to end QoS that can be achieved when IntServ runs over the DiffServ network in the next generation Internet. Our *approach* is to add a mapping function to the edge DiffServ router so that the traffic flows coming from IntServ domain can be appropriately mapped into the corresponding *Behavior Aggregates* of DiffServ, and then marked with the appropriate DSCP (Differentiated Service Code Point) for routing in the DiffServ domain. We show that, without making any significant changes to the IntServ or DiffServ infrastructure and without any additional protocols or signaling, it is possible to provide QoS to IntServ applications when IntServ runs over a DiffServ network.

The *significance* of this work is that end-to-end QoS over heterogeneous networks could be possible if the DiffServ backbone is used to connect IntServ subnetworks in the next generation Internet. The main *contributions* of this paper can be summarized as follows:

- Propose a mapping function to run IntServ over the DiffServ backbone.
- Show that QoS can be achieved by end IntServ applications when running over DiffServ backbone in the next generation Internet.

The rest of this paper is organized as follows. In Sections 2 and 3, we briefly present the main features of IntServ and DiffServ, respectively. In Section 4, we describe our approach for the mapping from IntServ to DiffServ and the simulation configuration to test the effectiveness of our approach. In Section 5, we analyze our simulation results to show that QoS can be provided to end applications in the IntServ domain. Concluding remarks are finally given in Section 6.

2 Integrated Services

The Integrated Services (IntServ) model [1] characterized by resource reservation defines a set of extensions to the traditional *best effort* model with the goal of providing end-to-end QoS to applications. This architecture needs some explicit signaling mechanism to convey information to routers so that they can provide requested services to flows that require them. RSVP is one of the most widely known example of such a signaling mechanism. We will describe this mechanism in details in Section 2.2. In addition to the *best effort* service, the integrated services model provides two service levels as follows.

- *Guaranteed service* [7] for applications requiring firm bounds on end-to-end datagram queueing delays.

- *Controlled-load service* [8] for applications requiring services closely equivalent to that provided to uncontrolled *best effort* traffic under unloaded (lightly loaded) network conditions.

We will discuss them in Sections 2.3 and 2.4, respectively.

2.1 Components of Integrated Services

The basic framework of integrated services [4] is implemented by four components: the *signaling protocol* (e.g., RSVP), the *admission control routine*, the *classifier* and the *packet scheduler*. In this model, applications must set up paths and reserve resources before transmitting their data. Network

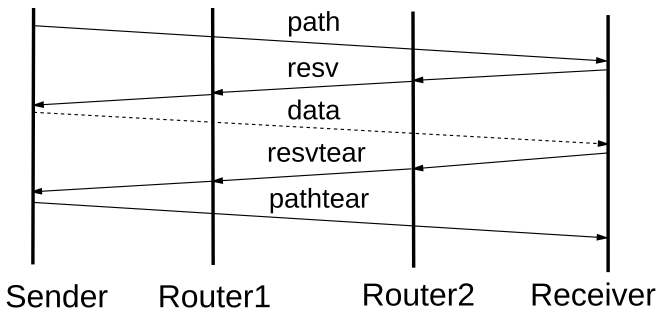


Figure 1: RSVP signaling for resource reservation.

elements will apply admission control to those requests. In addition, traffic control mechanisms on the network element are configured to ensure that each admitted flow receives the service requested in strict isolation from other traffic. When a router receives a packet, the classifier will perform a MF (multifield) classification and put the packet in a specific queue. The packet scheduler will then schedule the packet according to its QoS requirements.

2.2 RSVP Signaling

RSVP is a signaling protocol to reserve network resources for applications. Figure 1 illustrates the setup and teardown procedures of PSVP protocol. The sender sends a **PATH** message to the receiver specifying the characteristic of the required traffic. Every intermediate router along the path forwards the **PATH** message to the next hop determined by the routing protocol. If the receiver agrees the advertised flow, it sends a **RESV** message, which is forwarded hop by hop via RSVP capable routers towards the sender of the **PATH** message. Every intermediate router along the path may reject or accept the request. If the request is accepted, resources are allocated, and **RESV** message is forwarded. If the request is rejected, the router will send an **RESV-ERR** message back to the sender of the **RESV** message.

If the sender gets the **RESV** message, it means resources are reserved and data can be transmitted. To terminate a reservation, a **RESV-TEAR** message is transmitted to remove the resource allocation and a **PATH-TEAR** message is sent to delete the path states in every router along the path.

2.3 Guaranteed Service

Guaranteed service guarantees that datagrams will arrive within the guaranteed delivery time and will not be discarded due to queue overflows, provided the flow's traffic stays within its specified traffic parameters [7]. The service provides assured level of bandwidth or link capacity for the data flow. It imposes a strict upper bound on the end-to-end queueing delay as data flows through the network. The packets encounter no queueing delay as long as they conform to the flow specifications. It means packets cannot be dropped due to buffer overflow and they are always guaranteed the required buffer space. The delay bound is usually large enough even to accommodate cases of long queueing delays.

2.4 Controlled-load Service

The controlled-load service does not accept or make use of specific target values for control parameters such as delay or loss. Instead, acceptance of a request for controlled-load service is defined to imply a commitment by the network elements to provide the requester with a service closely equivalent to that provided to uncontrolled (best effort) traffic under lightly loaded conditions [8]. The service aims at providing the same QoS under heavy loads as under unloaded conditions. Though there is no specified strict bound on delay, it ensures that very high percentage of packets do not experience delays highly greater than the minimum transit delay due to propagation and router processing.

3 Differentiated Services

The IntServ/RSVP architecture described in Section 2 can be used to provide QoS to applications. All the routers are required to be capable of RSVP, admission control, MF classification and packet scheduling, which needs to maintain all the information for each flow at each router. The above issues raise scalability concerns in large networks [4]. Because of the difficulty in implementing and deploying integrated services and RSVP, differentiated services is currently being developed by the IETF [2].

Differentiated services (DiffServ) is intended to enable the deployment of scalable service discrimination in the Internet without the need for per-flow state and signaling at every hop. The premise of DiffServ networks is that routers in the core network handle packets from different traffic streams by forwarding them using different per-hop behaviors (PHBs). The PHB to be applied is indicated by a DiffServ Codepoint (DSCP) in the IP header of the packet [9]. The advantage of such a mechanism is that several different traffic streams can be aggregated to one of a small number of behavior aggregates (BA) which are each forwarded using the same PHB at the router, thereby simplifying the processing and associated storage [10]. There is no signaling or processing since QoS (Quality of Service) is invoked on a packet-by-packet basis [10].

The DiffServ architecture is composed of a number of functional elements, including a small set of per-hop forwarding behaviors, packet classification functions, and traffic conditioning functions which includes metering, marking, shaping and policing. The functional block diagram of a typical DiffServ router is shown in Figure 2 [10]. This architecture provides *Expedited Forwarding* (EF) service and *Assured Forwarding* (AF) service in addition to *best-effort* (BE) service as described below.

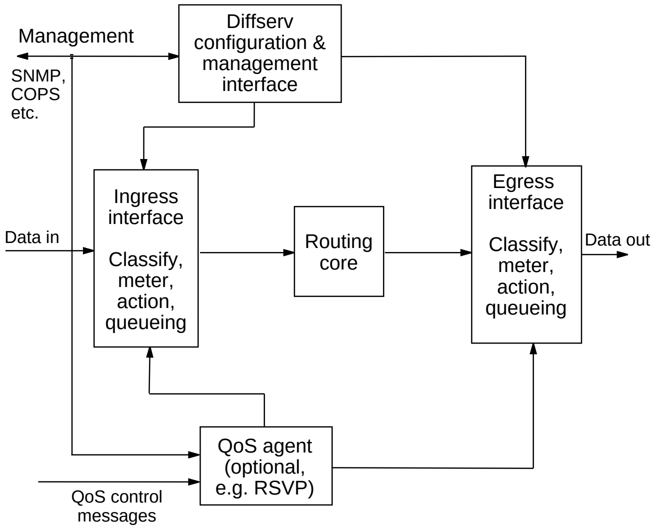


Figure 2: Major functional block diagram of a router.

3.1 Expedited Forwarding (EF)

This service is also been described as *Premium Service*. The EF service provides a low loss, low latency, low jitter, assured bandwidth, end-to-end service for customers [11]. Loss, latency and jitter are due to the queuing experienced by traffic while transiting the network. Therefore, providing low loss, latency and jitter for some traffic aggregate means there are no queues (or very small queues) for the traffic aggregate. At every transit node, the aggregate of the EF traffic's maximum arrival rate must be less than its configured minimum departure rate so that there is almost no queuing delay for these premium packets. Packets exceeding the peak rate are shaped by the traffic conditioners to bring the traffic into conformance.

3.2 Assured Forwarding

This service provides a reliable services for customers, even in times of network congestion. Classification and policing are first done at the edge routers of the DiffServ network. The assured service traffic is considered *in*-profile if the traffic does not exceed the bit rate allocated for the service; otherwise, the excess packets are considered *out*-of-profile. The *in*-profile packets should be forwarded

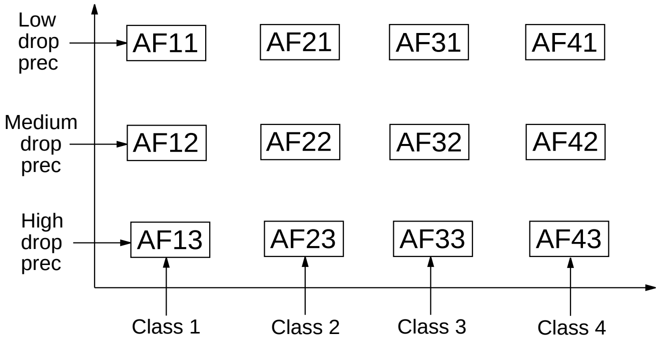


Figure 3: AF classes with drop precedence levels.

with high probability. However, the *out*-of-profile packets are not delivered with as high probability as the traffic that is within the profile. Since the network does not reorder packets that belong to the same microflow, all packets, irrespective of whether they are *in*-profile or *out*-of-profile, are put into an *assured queue* to avoid out-of-order delivery.

Assured Forwarding provides the delivery of packets in four independently forwarded AF classes. Each class is allocated with a configurable minimum amount of buffer space and bandwidth. Each class is in turn divided into different levels of drop precedence. In the case of network congestion, the drop precedence determines the relative importance of the packets within the AF class. Figure 3 [12] shows four different AF classes with three levels of drop precedence.

3.3 Best Effort

This is the default service available in DiffServ, and is also deployed by the current Internet. It does not guarantee any bandwidth to the customers, but can only get the bandwidth available. Packets are queued when buffers are available and dropped when resources are over committed.

4 Integrated Services over Differentiated Services Networks

In this section, we describe in details the mapping strategy adopted in this paper to connect the IntServ and DiffServ domains. Simulation configuration that has been used to test the mapping strategy is described in 4.3 .

4.1 Mapping Considerations for IntServ over DiffServ

In IntServ, resource reservations are made by requesting a service type specified by a set of quantitative parameters known as *Tspec* (Traffic Specification). Each set of parameters determines an appropriate priority level. When requested services with these priority levels are mapped to DiffServ domain, some basic requirements should be satisfied.

- PHBs in DiffServ domain must be appropriately selected for each requested service in IntServ domain.
- The required policing, shaping and marking must be done at the edge router of the DiffServ domain.
- Taking into account the resource availability in DiffServ domain, admission control must be implemented for requested traffic in IntServ domain.

4.2 Mapping Function

The mapping function is used to assign an appropriate DSCP to a flow specified by *Tspec* parameters in IntServ domain, such that the same QoS could be achieved for IntServ when running over DiffServ domain. Each packet in the flow from the IntServ domain has a *flow ID* indicated by the value of *flow-id* field in the IP (Internet Protocol) header. The *flow ID* attributed with the *Tspec* parameters is used to determine which flow the packet belongs to. The main constraint is that the

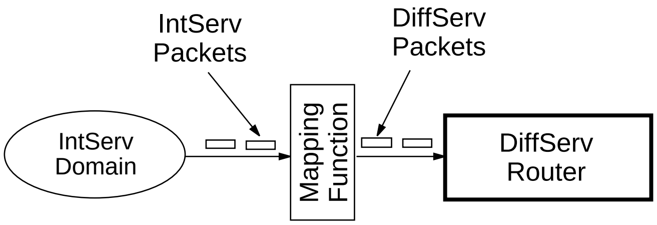


Figure 4: Mapping function for integrated service over differentiated service.

PHB treatment of packets along the path in the DiffServ domain must approximate the QoS offered by IntServ itself. In this paper, we satisfy the above requirement by appropriately mapping the flows coming from IntServ domain into the corresponding *Behavior Aggregates*, and then marking the packets with the appropriate DSCP for routing in the DiffServ domain.

To achieve the above goal, we introduce a mapping function at the boundary router in DiffServ domain as shown in Figure 4. Packets specified by *Tspec* parameters in IntServ domain are first mapped to the corresponding PHBs in the DiffServ domain by appropriately assigning a DSCP according to the mapping function. The packets are then routed in the DiffServ domain where they receive treatment based on their DSCP code. The packets are grouped to BAs in the DiffServ domain. Table 1 shows an example mapping function which has been used in our simulation. As an instance, a flow in IntServ domain specified by *r=0.7Mb, b=5000bytes* and *Flow ID=0* is mapped to *EF* PHB (with corresponding DSCP 101110) in DiffServ domain, where *r* means token bucket rate and *b* means token bucket depth.

Table 1: An example mapping function used in our simulation.

Tspec	*Flow ID*	*PHB*	*DSCP*
r=0.7 Mb, b=5000 bytes	0	EF	101110
r=0.7 Mb, b=5000 bytes	1	EF	101110
r=0.5 Mb, b=8000 bytes	2	AF11	001010
r=0.5 Mb, b=8000 bytes	3	AF11	001010
r=0.5 Mb, b=8000 bytes	4	AF11	001010

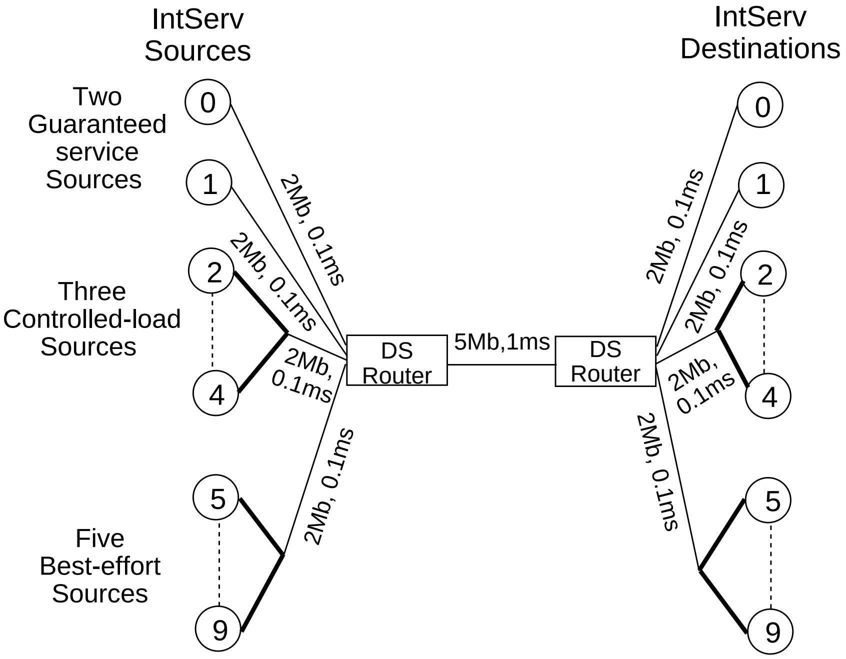


Figure 5: Network simulation configuration.

The sender initially specifies its requested service using *Tspec*. Note that it is possible for different senders to use the same *Tspec*. However, they are differentiated by the *flow ID*. In addition, it is also possible that different flows can be mapped to the same PHB in DiffServ domain.

4.3 Simulation Configuration

To test the effectiveness of our proposed mapping strategy between IntServ and DiffServ and to determine the QoS that can be provided to IntServ applications, we carried out simulation using the *ns* (Version 2.1b6) simulation tool from Berkeley [13]. The network configuration used in our simulation is shown in Figure 5.

Ten IntServ sources were used in our simulation, the number of sources generating *Guaranteed services*, *Controlled-load* services and *best-effort* services were two, three and five respectively. Ten IntServ sinks served as destinations for the IntServ sources. We set the *flow IDs* to be the same as the corresponding source number shown in Figure 5.

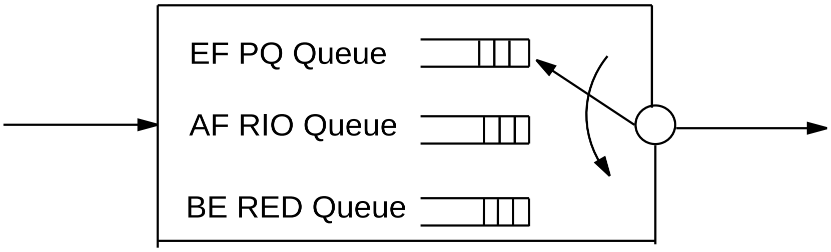


Figure 6: Queues inside the edge DiffServ router.

All the links in Figure 5 are labeled with a (*bandwidth, propagation delay*) pair. The mapping function shown in Table 1 has been integrated into the DiffServ edge router (See Figure 4). CBR (Constant Bit Rate) traffic was used for all IntServ sources in our simulation so that the relationship between the bandwidth utilization and bandwidth allocation can be more easily evaluated. *Note that ten admission control modules have been applied to each link between sources and DiffServ edge routers to guarantee the resource availability within DiffServ domain. To save space, they are not illustrated in Figure 5. Admission control algorithm was implemented by token bucket with parameters specified in Table 1.*

Inside the DiffServ edge router, EF queue was configured as a simple Priority Queue with Tail Drop; AF queue was configured as RIO queue and BE queue as a RED [14] queue, which are shown in Figure 6. The queue weights of EF, AF and BE queues were set to 0.4, 0.4 and 0.2 respectively. Since the bandwidth of the bottleneck link between two DiffServ routers is 5 Mb, the above scheduling weights implies bandwidth allocations of 2 Mb, 2 Mb and 1 Mb for the EF, AF and BE links respectively during periods of congestion at the edge router.

5 Simulation Results

In this section, results obtained from our simulation experiments are presented. The criteria used to evaluate our proposed strategy are first described followed by the explanations of our experimental and numerical results.

5.1 Performance Criteria

To show the effectiveness of our mapping strategy in providing QoS to end IntServ applications, we have used *goodput*, *queue size* and *drop ratio* as the performance criteria. In addition, in order to prove the effectiveness of admission control mechanism, we also measured the *non-conformant ratio* (the ratio of non-conformant packets out of in-profile packets). In Section 5.2, we present the results of measurements of the above quantities from our simulation experiments.

5.2 QoS Obtained by Guaranteed Services

We use the following three simulation cases to determine the QoS obtained by IntServ applications. As results, Table 2 shows the goodput of each *Guaranteed service* source for three different cases described in Section 5.2. Table 3 shows the drop ratio measured at the scheduler for three cases of the *Guaranteed service* sources. Table 4 shows the non-conformant ratio for each *Guaranteed service* source. Figures 7, 8 and 9 show the queue size for each of the three case, from which the queuing delay and jitter can be evaluated.

Table 2: Goodput of each *Guaranteed service* source (Unit: Kb/S)

Tspec	*Flow ID*	*Case 1*	*Case 2*	*Case 3*
r=0.7 Mb, b=5000 bytes	*0*	699.8250	699.8039	459.8790
r=0.7 Mb, b=5000 bytes	*1*	699.8039	699.6359	1540.1400

Table 3: Drop ratio of *Guaranteed service* traffic.

Type of traffic	*Case 1*	*Case 2*	*Case 3*
Guaranteed service Traffic	0.000000	0.000000	0.258934

Table 4: The non-conformant ratio for each *Guaranteed service* source

Tspec	*Flow ID*	*Case 1*	*Case 2*	*Case 3*
r=0.7 Mb, b=5000 bytes	*0*	0.00026	0.00026	0.00026
r=0.7 Mb, b=5000 bytes	*1*	0.00026	0.22258	0.00040

5.2.1 Case 1: No congestion; no excessive traffic

The traffic generated by *Guaranteed service* sources (source *0* and source *1*) were set to 0.7 Mb and 0.7 Mb, respectively. In this case, the traffic rate is equal to the bucket rate (0.7 Mb, shown in Table 1), which means *there should not be any significant excessive IntServ traffic.* According to the network configuration described in Section 4.3, two *Guaranteed service* sources generate 1.4 Mb traffic which is less than the corresponding scheduled link bandwidth for *Guaranteed service* (EF in DiffServ domain) traffic (2Mb). Under this scenario, *there should not be any significant congestion* at the edge DiffServ router.

Case 1 is an ideal case. As seen in Table 2, the goodput is almost equal to the corresponding source rate. From Table 3, since there is no significant congestion, the drop ratio of each type of sources is zero. Table 4 shows the performance of admission control mechanism. Since there is no excessive traffic in this case, the non-conformant ratio is almost zero. Figure 7 shows the queuing performance of each queue. Because this is an ideal case, the size of each queue is very small. Though the three queues have almost the same average size, we observe that the BE queue of IntServ (mapping to BE queue in DiffServ domain, according to the mapping function) has the largest jitter.

5.2.2 Case 2: No congestion; Guaranteed service source *1* generates excessive traffic

The traffic generated by *Guaranteed service* sources (source *0* and source *1*) were set to 0.7 Mb and 0.9 Mb, respectively. In this case, the traffic rate of source *1* is greater than its corresponding bucket

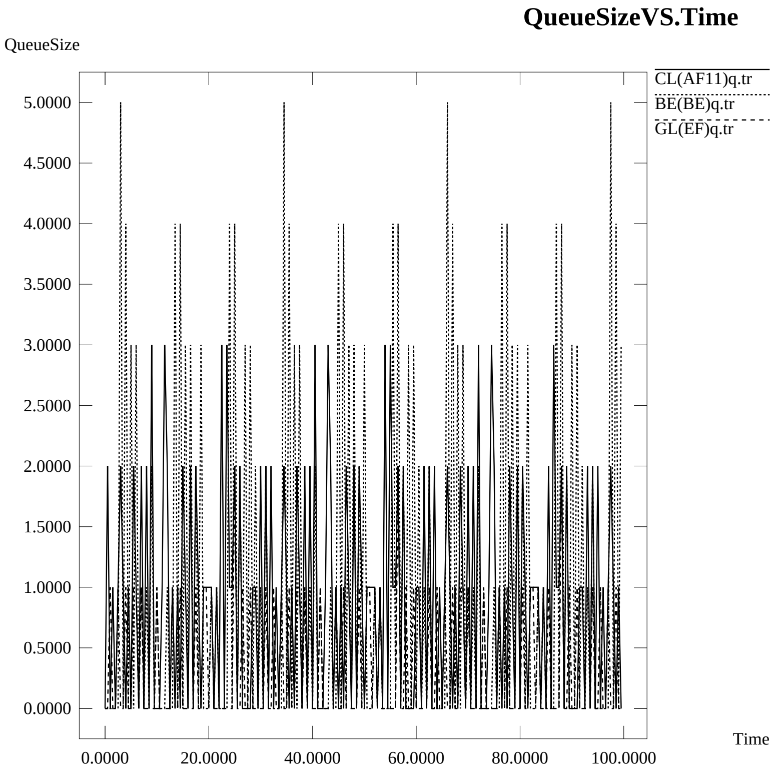


Figure 7: Queue size plots for *Case 1*.

rate (0.7 Mb, shown in Table 1), which means *source* 1 *generates excessive IntServ traffic*. According to the network configuration described in Section 4.3, two *Guaranteed service* sources generate 1.6 Mb traffic which is less than the corresponding scheduled link bandwidth for *Guaranteed service* (EF in DiffServ domain) traffic (2Mb). Under this scenario, *there should not be any significant congestion* at the edge DiffServ router.

In *case 2*, from Table 2, the goodput of source *0* is equal to its source rate. However, the goodput of source *1* is equal to the corresponding token rate, 0.7 Mb, rather than its source rate, 0.9 Mb. Table 3 shows that the drop ratio of *Guaranteed service* is *0*. The reason is that, in this case, there is no congestion for *Guaranteed service* traffic. Table 4 indicates how the admission control mechanism works. As seen in this table, the non-conformant packets ratio of source *1* is increased, compared to case 1. It is because source *1* generates excessive traffic in this case. From Figure 8, we find that the average queue size of the *best effort* queue is far greater than the other

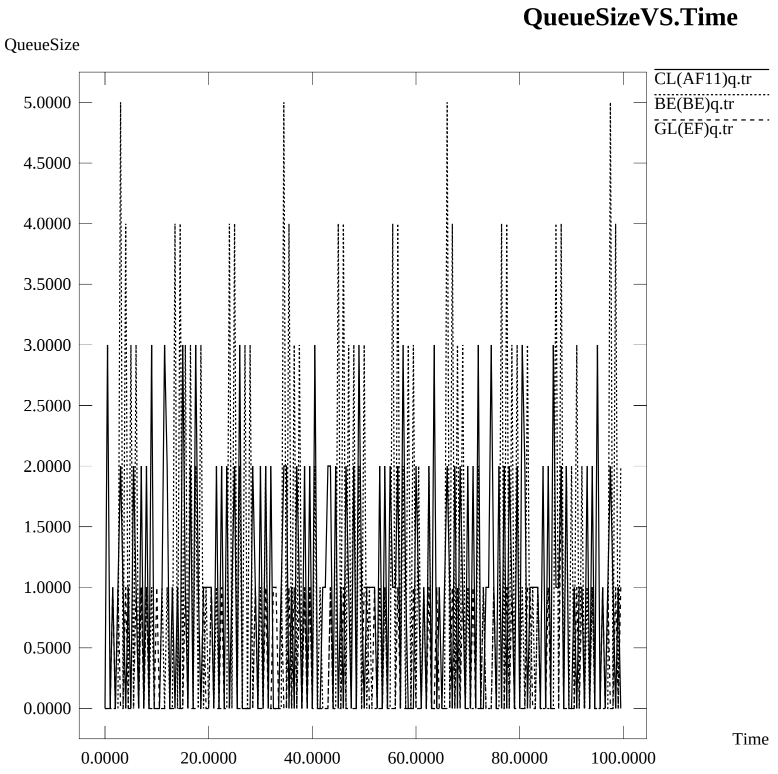


Figure 8: Queue size plots for *Case 2*.

two types of sources. In addition, the jitter of *best effort* traffic is also greater than the other two types of sources. The *Guaranteed service* traffic has the smallest average queue size and the smallest jitter. In addition, compared with Figure 7, the upper bound of *Guaranteed service* queue is guaranteed, though the source *1* generates more traffic than what it has reserved. This well satisfies requirements from [7].

5.2.3 Case 3: Guaranteed service gets into congestion; no excessive traffic

The traffic generated by *Guaranteed service* sources (source *0* and source *1*) were set to 0.7 Mb and 2 Mb, respectively. To simulate a congested environment, we set the token rate of source *1* to 2 Mb also. In this case, the traffic rate of source *1* is equal to its corresponding bucket rate (2 Mb), which means *there is no significant excessive IntServ traffic.* According to the network configuration described in Section 4.3, two *Guaranteed service* sources generate 2.7 Mb traffic which

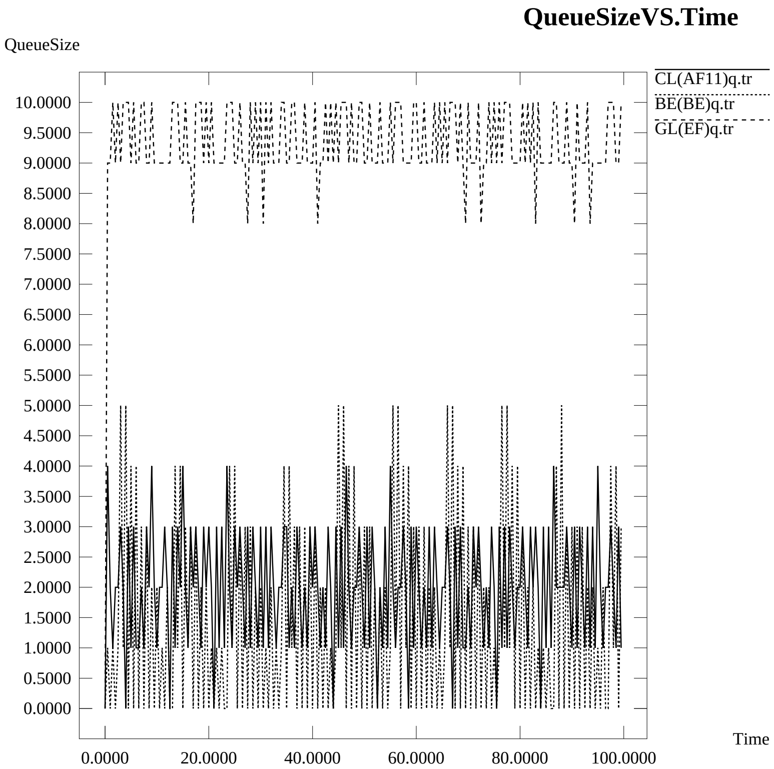


Figure 9: Queue size plots for *Case 3*.

is greater than the corresponding scheduled link bandwidth for *Guaranteed service* (EF in DiffServ domain) traffic (2Mb). Under this scenario, *Guaranteed service traffic gets into congestion* at the edge DiffServ router.

Case 3 is used to evaluate our mapping function under congested environments. As expected, we find the drop ratio (measured at scheduler) of *Guaranteed service* traffic is increased, and the total goodput of *Guaranteed service* is limited by the output link bandwidth assigned by the scheduler (2Mb), instead of 2.7 Mb. Since there is no excessive traffic, from Table 4, the no-conformant packets ratio of both of the *Guaranteed service* sources are closed to *0*. From Figure 9, since we increase the token rate of one of the *Guaranteed service* source (source *1*), the upper bound of *Guaranteed service* is increased, which is reasonable. In addition, the *Guaranteed service* queue still has the smallest jitter.

5.3 QoS Obtained by Controlled-load Services

Because of the similarity between the results of *Guaranteed service* and *Controlled-load service*, *all our descriptions in Section 5.2 are focused on* Guaranteed service. *We only give out results for Controlled-load service without detailed explanations.*

We use *case 2* described in Section 5.2.2 as an example. As described in Section 4.3, we used three *Controlled-load service* sources in our simulation: sources *2*, *3* and *4*. The token bucket parameters are shown in Table 1. We set the source rate of sources *2* and *4* to 0.5 Mb, 0.5 Mb, respectively, and set the rate of source *3* to 0.7 Mb (greater than its token rate, 0.5 Mb). Therefore, *source* 3 *generates excessive traffic*. The total *Controlled-load service* traffic is 1.7 Mb, which is less than the scheduled link bandwidth; therefore, *there should not be any significant congestion.*

Table 5 shows the goodput of each *Controlled-load* source. Table 6 shows the drop ratio of *Controlled-load service* measured at scheduler. Table 7 shows the non-conformant ratio. Figure 10 shows the queue size of this case. Note that though the non-conformant ratio of source *3* is much higher that the other two (shown in Table 7), the goodput of source*3* (shown in Table 5) is equal to its source rate (0.7 Mb). It is because the non-conformant packets are degraded and then forwarded, which is one of the forwarding schemes for non-conformant packets proposed by [8].

Table 5: Goodput of each *Controlled-load service* source (Unit: Kb/S)

Tspec	*Flow ID*	*Case 2*
r=0.5 Mb, b=8000 bytes	*2*	499.9889
r=0.5 Mb, b=8000 bytes	*3*	700.0140
r=0.5 Mb, b=8000 bytes	*4*	499.9889

Table 6: Drop ratio of *Controlled-load service* traffic.

Type of traffic	*Case 2*
Controlled-load Traffic	0.000000

Table 7: The non-conformant ratio for each *Controlled-load service* source

Tspec	*Flow ID*	*Case 2*
r=0.5 Mb, b=8000 bytes	*2*	0.00000
r=0.5 Mb, b=8000 bytes	*3*	0.28593
r=0.5 Mb, b=8000 bytes	*4*	0.00000

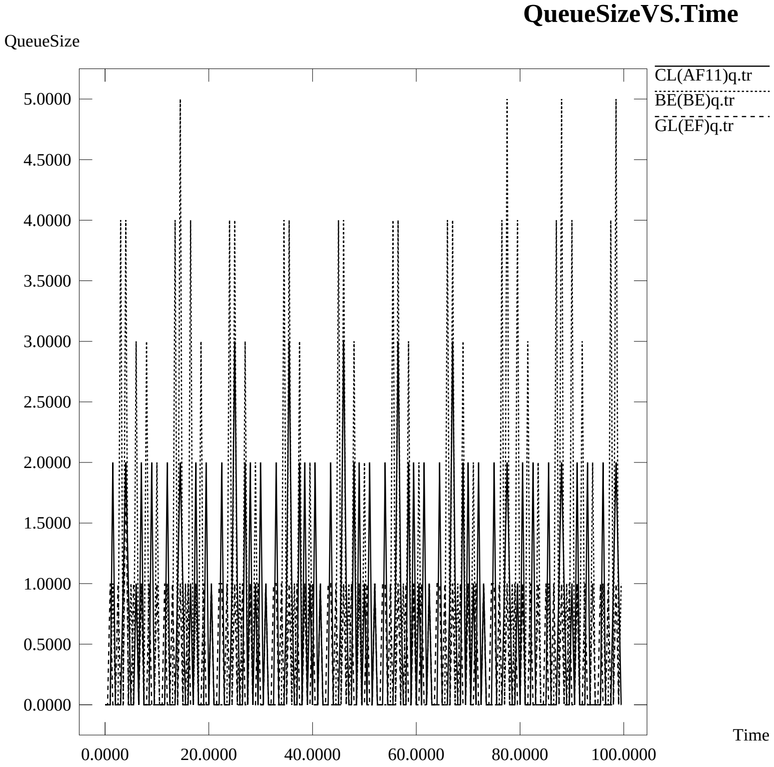


Figure 10: Queue size plots.

5.4 Observations

From the above results, we can arrive at the following *observations*:

- The upper bound of queueing delay of *Guaranteed service* is guaranteed. In addition, *Guaranteed service* always has the smallest jitter without being affected by other traffic flows, though [7] says it does not attempt to minimize the jitter. This well satisfies requirements from [7].

- The *Controlled-load service* has the smaller jitter and queue size than the *best effort* traffic. Furthermore, non-conformant packets are degraded and then forwarded, which is proved by our simulation. This well satisfies requirements from [8].

We therefore, *conclude that the QoS requirements of IntServ can be successfully achieved when IntServ traffic is mapped to the DiffServ domain in next generation Internet.*

6 Conclusion

In this paper, we have proposed DiffServ as the backbone network to interconnect IntServ subnetworks. We have designed a mapping function to map traffic flows coming from IntServ with different priorities to the corresponding PHBs in the DiffServ domain.

The proposed scheme has been studied in detail using simulation. It has been found that the QoS requirements of IntServ can be achieved when IntServ subnetworks run over DiffServ. We have illustrated our scheme by mapping IntServ traffic of three different priorities to the three service classes of DiffServ. The ability of our scheme to provide QoS to end IntServ applications has been demonstrated by measuring the *drop ratio*, *goodput*, *non-conformant ratio* and *queue size*. We found that the upper bound of queueing delay of *Guaranteed service* is guaranteed. In addition, *Guaranteed service* always has the smallest jitter without being affected by other traffic flows, though [7] says it does not attempt to minimize the jitter. The *Controlled-load service* has the smaller jitter and queue size than the *best effort* traffic. Furthermore, non-conformant packets are degraded and then forwarded, which is proved by our simulation.

Acknowledgements

The authors thank NASA Glenn Research Center for support of this project. Jyotsna Chitri carried out some initial work on this project.

References

[1] R. Braden, D. Clark, and S. Shenker, "Integrated services in the internet architecture: an overview." RFC 1633, June 1994.

[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services." RFC 2475, December 1998.

[3] J. Wroclawski and A. Charny, "Integrated service mapping for differentiated services networks." draft-ietf-issll-ds-map-00.txt, March 2000.

[4] Y. Bernet, R. Yavatkar, P. Ford, F. Baker, L. Zhang, M. Speer, R. Braden, B. Davie, J. Wroclawski, and E. Felstaine, "A framework for integrated services operation over diffserv networks." draft-ietf-issll-diffserv-rsvp-04.txt, March 2000.

[5] R. Balmer, F. Baumgarter, and T. Braun, "A concept for RSVP over diffserv," *Proc. Nineth ICCCN'2000*, Las Vagas, NV, October 2000.

[6] J. Harju and P. Kivimäki, "Co-operation and comparison of diffserv and intserv: Performance measurements," *Proc. 25th Conference on Local Computer Networks*, Tampa, FL, pp. 177–186, November 2000.

[7] S. Shenker, C. Partridge, and R. Guerin, "Specification of guaranteed quality of service." RFC 2212, September 1997.

[8] J. Wroclawski, "Specification of the controlled-load network element service." RFC 2211, September 1997.

[9] K. Nichols, S.Blake, F. Baker, and D.Black, "Definition of the differentiated services field (DS field) in the IPv4 and IPv6 headers." RFC 2474, December 1998.

[10] Y. Bernet, S. Blake, D. Grossman, and A. Smith, "An informal management model for diffserv routers." draft-ietf-diffserv-model-04.txt, July 2000.

[11] V. Jacobson, K. Nichols, and K. Poduri, "An expedited forwarding PHB." RFC 2598, June 1999.

[12] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured forwarding PHB group." RFC 2597, June 1999.

[13] VINT Project U.C. Berkeley/LBNL, "ns v2.1b6: Network simulator." http://www-mash.cs.berkeley.edu/ns/, January 2000.

[14] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, August 1993.

Paper 3

H. Bai, M. Atiquzzaman and W. Ivancic, “Achieving QoS for Aeronautical Telecommunication Networks over Differentiated Services”

Achieving QoS for Aeronautical Telecommunication Networks over Differentiated Services [1]

Haowei Bai and Mohammed Atiquzzaman [2]
Department of Electrical and Computer Engineering
The University of Dayton, Dayton, Ohio 45469-0226
E-mail: baihaowe@flyernet.udayton.edu, atiq@ieee.org
William Ivancic
NASA Glenn Research Center
21000 Brookpark Rd. MS 54-8,
Cleveland, OH 44135

Abstract

Aeronautical Telecommunication Network (ATN) has been developed by the International Civil Aviation Organization to integrate Air-Ground and Ground-Ground data communication for aeronautical applications into a single network serving Air Traffic Control and Aeronautical Operational Communications [1]. To carry time critical information required for aeronautical applications, ATN provides different Quality of Services (QoS) to applications. ATN has therefore, been designed as a standalone network which implies building an expensive separate network for ATN. However, the cost of operating ATN can be reduced if it can be run over a public network such as the Internet. Although the current Internet does not provide QoS, the next generation Internet is expected to provide QoS to applications. The *objective* of this paper is to investigate the possibility of providing QoS to ATN applications when it is run over the next generation Internet. Differentiated Services (DiffServ), one of the protocols proposed for the next generation Internet, will allow network service providers to offer different QoS to customers. Out results show that it is possible to provide QoS to ATN applications when they run over a DiffServ backbone.

1 Introduction

The International Civil Aviation Organization (ICAO) has developed the Aeronautical Telecommunication Network (ATN) as a commercial infrastructure to integrate Air-Ground and Ground-Ground data communication into a single network to serve air traffic control and aeronautical operational communications [1]. One of the objectives of ATN internetwork is to accommodate different Quality of Service (QoS) required by ATSC (Air Traffic Services Communication) and AINSC (Aeronautical Industry Service Communication) applications, and the organizational policies for interconnection and routing specified by each participating organization. In the ATN, priority has the essential role of ensuring that high priority safety related and time critical data are not delayed by low priority non-safety data, especially when the network is overloaded with low priority data.

The time critical information carried by ATN and the QoS required by ATN applications has led to the development of the ATN as an expensive independent network. The largest public network, the Internet, only offers point-to-point *best-effort* service to the users and hence is not suitable for carrying time critical ATN traffic. However, the rapid commercialization of the Internet has given rise to demands for QoS over the Internet.

QoS is generally implemented by different classes of service contracts for different users. A service class may provide low-delay and low-jitter services for customers who are willing to pay

[1] The work reported in this project was supported by NASA grant no. NAG3-2318

[2] The second author is currently with the School of Computer Science, University of Oklahoma, Norman, OK 73072, Tel: (405) 325 8077, email: atiq@ou.edu

a premium price to run high-quality applications, such as, real-time multimedia. Another service class may provide predictable services for customers who are willing to pay for reliability. Finally, the *best-effort* service provided by current Internet will remain for those customers who need only connectivity.

The Internet Engineering Task Force (IETF) has proposed a few models to meet the demand for QoS. Notable among them are the Integrated Services (IntServ) model [2] and Differentiated Services (DiffServ) [3] model. The IntServ model is characterized by resource reservation; before data is transmitted, applications must set up paths and reserve resources along the path. This gives rise to scalability issues in the core routers of large networks. The DiffServ model is currently being standardized to overcome the above scalability issue, and to accommodate the various service guarantees required for time critical applications. The DiffServ model utilizes six bits in the TOS (Type of Service) field of the IP header to mark a packet for being eligible for a particular forwarding behavior. It The model does not require significant changes to the existing infrastructure, and does not need too many additional protocols.

A significant cost saving can be achieved if the ATN protocol could be run over the next generation Internet protocol as shown in Figure 1. In this paper, we are interested in developing a framework to run ATN over the next generation Internet. This requires appropriate mapping of parameters at the edge routers between the two networks. The *objective* of this paper is to investigate the QoS that can be achieved when ATN runs over the DiffServ network in the next generation Internet. Based on the similarity between an IP packet and an ATN packet, our *approach* is to add a mapping function to the edge DiffServ router so that the traffic flows coming from ATN can be appropriately mapped into the corresponding *Behavior Aggregates* of DiffServ, and then marked with the appropriate DSCP (Differentiated Service Code Point) for routing in DiffServ domain. We show that, without making any significant changes to the ATN or DiffServ infrastructure and without any additional protocols or signaling, it is possible to provide QoS to ATN applications when ATN runs over a DiffServ network.

The *significance* of this work is that considerable cost savings could be possible if the next generation Internet backbone can be used to connect ATN subnetworks. The main *contributions* of this paper can be summarized as follows:

- Propose a framework to run ATN over the DiffServ network.
- Show that QoS can be achieved by end ATN applications when run over the next generation Internet.

The rest of this paper is organized as follows. In Sections 2 and 3, we briefly present the main features of ATN and DiffServ, respectively. In Section 4, we describe our approach for the interconnection of ATN and DiffServ and the simulation configuration to test the effectiveness of our approach. In Section 5, we analyze our simulation results to show that QoS can be provided to end applications in the ATN domain. Concluding remarks are finally given in Section 6.

2 Aeronautical Telecommunication Network (ATN)

In the early 1980s, the International Civil Aviation Organization (ICAO) recognized the increasing limitations of the present air navigation systems and the need for improvements to take civil aviation into the 21st century. The need for changes in the current global air navigation system is due to two principal factors:

- The present and growing air traffic demand which the current system will be unable to cope.

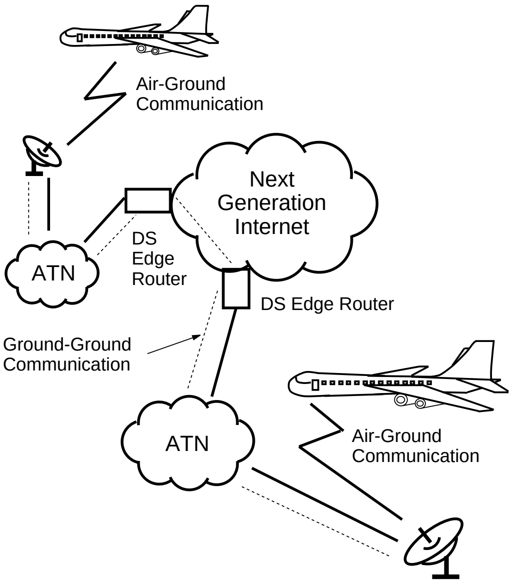


Figure 1: Interconnection between ATN and Differentiated Services.

- The need for global consistency in the provisioning of air traffic services during the progression towards a seamless air traffic management system.

The above factors gave rise to the concept of the Aeronautical Telecommunication Network (ATN) [4].

ATN is both a ground-based network providing communications between ground-based users, and an air-ground network providing communications between airborne and ground users. It was always intended that ATN should be built on existing technologies instead of inventing new approaches. The Internet approach was seen as the most suitable approach, and was therefore selected as the basis for the ATN. ATN is made up of End Systems, Intermediate Systems, ground-ground subnetworks and air-ground subnetworks as shown in Figure 1.

2.1 Priority in ATN

The ATN has been designed to provide a high reliability/availability network by ensuring that there is no single point of failure, and by permitting the availability of multiple alternative routes to the same destination with dynamic switching between alternatives. Every ATN user data is given a relative priority on the network in order to ensure that low priority data does not impede the flow of high priority data. The purpose of priority is to signal the relative importance and (or) precedence of data, such that when a decision has to be made as to which data to act first, or when contention for access to shared resources has to be resolved, the decision or outcome can be determined unambiguously and in line with user requirements both within and between applications.

Priority in ATN is signaled separately by the application in the transport layer, network layer, and in ATN subnetworks, which gives rise to *Transport Priority*, *Network Priority* and *Subnet Priority* [5]. Network priority is used to manage the access to network resources. During periods of high network utilization, higher priority NPDUs (Network Protocol Data Units) may therefore be expected to be more likely to reach their destination (i.e. be less likely to be discarded by a congested router), and to have a lower transit delay (i.e. be more likely to be selected for transmission from an outgoing queue) than lower priority packets. In this paper, we focus on *network priority* which

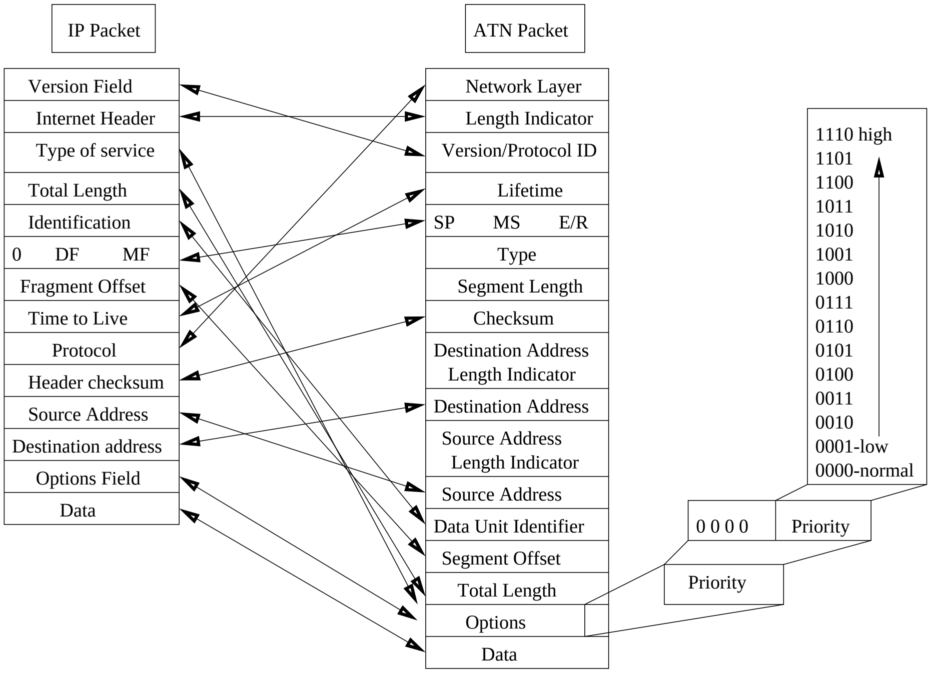


Figure 2: Similarity between an IP packet and an ATN packet

determines the sharing of limited network resources.

2.2 ATN packet format

Figure 2 shows the correspondence between the fields of an IP packet header and the network layer packet header of ATN. It is seen that the fields of IP and ATN packets carry similar information, and thus can almost be mapped to each other. This provides the possibility for mapping ATN to DiffServ (which uses the IP packet header except for the Type of Service byte) to achieve the required QoS when they are interconnected.

The NPDU header of an ATN packet contains an option part including an 8-bit field named *Priority* which indicates the relative priority of the NPDU [1]. The values 0000 0001 through 0000 1110 are to be used to indicate the priority in an increasing order. The value 0000 0000 indicates normal priority.

3 Differentiated Services

Differentiated services (Diffserv) is intended to enable the deployment of scalable service discrimination in the Internet without the need for per-flow state and signaling at every hop. The premise of Diffserv networks is that routers in the core network handle packets from different traffic streams by forwarding them using different per-hop behaviors (PHBs). The PHB to be applied is indicated by a Diffserv Codepoint (DSCP) in the IP header of the packet [6]. The advantage of such a mechanism is that several different traffic streams can be aggregated to one of a small number of behavior aggregates (BA) which are each forwarded using the same PHB at the router, thereby simplifying the processing and associated storage [7]. There is no signaling or processing since QoS (Quality of Service) is invoked on a packet-by-packet basis [7].

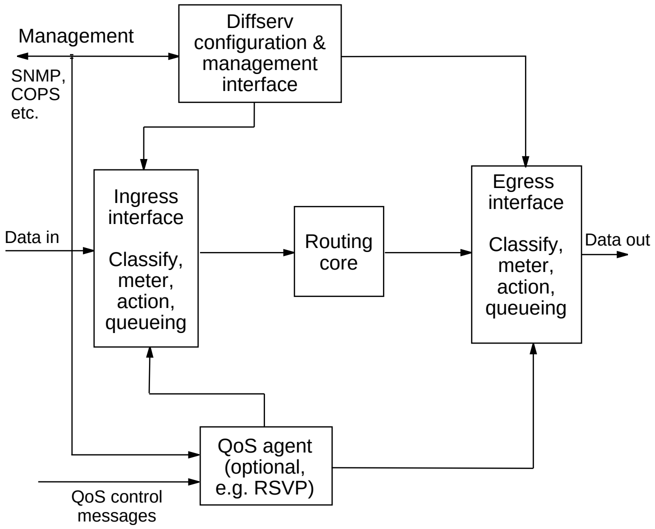

Management
SNMP,
COPS
etc.
Diffserv configuration & management interface
Ingress interface
Classify, meter, action, queueing
Data in
Routing core
Egress interface
Classify, meter, action, queueing
Data out
QoS agent (optional, e.g. RSVP)
QoS control messages

Figure 3: Major functional block diagram of a router.

The Diffserv architecture is composed of a number of functional elements, including a small set of per-hop forwarding behaviors, packet classification functions, and traffic conditioning functions which includes metering, marking, shaping and policing. The functional block diagram of a typical Diffserv router is shown in Figure 3 [7]. This architecture provides *Expedited Forwarding* (EF) service and *Assured Forwarding* (AF) service in addition to *best-effort* (BE) service as described below.

3.1 Expedited Forwarding (EF)

This service is also been described as *Premium Service*. The EF service provides a low loss, low latency, low jitter, assured bandwidth, end-to-end service for customers [8]. Loss, latency and jitter are due to the queuing experienced by traffic while transiting the network. Therefore, providing low loss, latency and jitter for some traffic aggregate means there are no queues (or very small queues) for the traffic aggregate. At every transit node, the aggregate of the EF traffic's maximum arrival rate must be less than its configured minimum departure rate so that there is almost no queuing delay for these premium packets. Packets exceeding the peak rate are shaped by the traffic conditioners to bring the traffic into conformance.

3.2 Assured Forwarding

This service provides a reliable services for customers, even in times of network congestion. Classification and policing are first done at the edge routers of the DiffServ network. The assured service traffic is considered *in*-profile if the traffic does not exceed the bit rate allocated for the service; otherwise, the excess packets are considered *out*-of-profile. The *in*-profile packets should be forwarded with high probability. However, the *out*-of-profile packets are not delivered with as high probability as the traffic that is within the profile. Since the network does not reorder packets that belong to the same microflow, all packets, irrespective of whether they are *in*-profile or *out*-of-profile, are put into an *assured queue* to avoid out-of-order delivery.

Assured Forwarding provides the delivery of packets in four independently forwarded AF classes. Each class is allocated with a configurable minimum amount of buffer space and bandwidth. Each

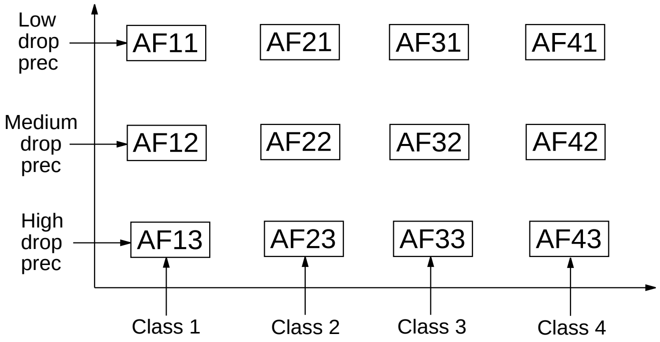


Figure 4: AF classes with drop precedence levels

class is in turn divided into different levels of drop precedence. In the case of network congestion, the drop precedence determines the relative importance of the packets within the AF class. Figure 4 [9] shows four different AF classes with three levels of drop precedence.

3.3 Best Effort

This is the default service available in DiffServ, and is also deployed by the current Internet. It does not guarantee any bandwidth to the customers, but can only get the bandwidth available. Packets are queued when buffers are available and dropped when resources are over committed.

4 ATN over Differentiated Services

In this section, we describe in detail the mapping strategy adopted in this paper to connect the ATN and DS domains followed by the simulation configuration we have used to test the mapping.

4.1 Mapping Function

Our goal is to use differentiated services to achieve QoS for ATN to integrate Air/Ground and Ground/Ground data communications into a global Internet serving Air Traffic Control (ATC) and Aeronautical Operations Communications (AOC). The main constraint is that the PHB treatment of packets along the path in the DiffServ domain must approximate the QoS offered in the ATN network. In this paper, we satisfy the above requirement by appropriately mapping the traffic coming from ATN into the corresponding *Behavior Aggregates*, and then marking the packets with the appropriate DSCP for routing in the DiffServ domain.

To achieve the above goal, we introduce a mapping function at the boundary router between the ATN and DiffServ domain as shown in Figure 5. Packets with different priorities from the ATN domain are first mapped to the corresponding PHBs in the DiffServ domain by appropriately assigning a DSCP according to the mapping function. The packets are then routed in the DiffServ domain where they receive treatment based on their DSCP code. The packets are grouped to BAs in the DiffServ domain. Table 1 shows an example mapping function which has been used in our simulation.

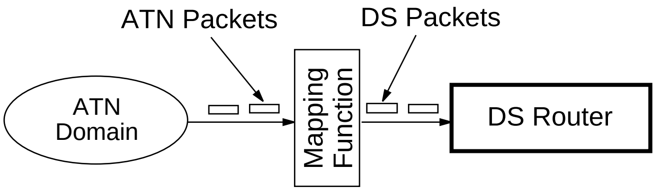


Figure 5: Mapping function for ATN over Differentiated Service.

Table 1: An example mapping function used in our simulation.

ATN Priority Code	*Priority*	*PHB*	*DSCP*
0000 0000	Normal	BE	000000
0000 0111	Medium	AF11	001010
0000 1110	High	EF	101110

4.2 Simulation Configuration

To test the effectiveness of our proposed mapping strategy between ATN and DiffServ and to determine the QoS that can be provided to ATN applications, we carried out simulation using the *ns* (Version 2.1b6) simulation tool from Berkeley [10]. The network configuration used in our simulation is shown in Figure 6.

Ten ATN sources were used in our simulation, the number of sources generating *high*, *medium* and *normal* priority packets were two, three and five respectively. Ten ATN sinks served as destinations for the ATN sources.

All the links in Figure 6 are labeled with a (*bandwidth, propagation delay*) pair. For the purpose of ATN over Diffserv, the mapping function shown in Table 1 has been integrated into the edge DiffServ router. CBR (Constant Bit Rate) traffic was used for all ATN sources in our simulation so that the relationship between the bandwidth utilization and bandwidth allocation can be more easily evaluated.

Inside the DiffServ router, EF queue was configured as a simple *Priority Queue* with *Tail Drop*. AF queue was configured as RIO queue and BE queue as a RED [11] queue. The queue weights of EF, AF and BE queues were set to 0.4, 0.4 and 0.2 respectively. Since the bandwidth of the bottleneck link between two DiffServ routers is 5 Mb, the above scheduling weights implies bandwidth allocations of 2 Mb, 2 Mb and 1 Mb for the EF, AF and BE links respectively during periods of congestion at the edge router.

5 Simulation Results

In this section, results obtained from our simulation experiments are presented. The criteria used to evaluate our proposed strategy are described followed by the description of our experiments and numerical results.

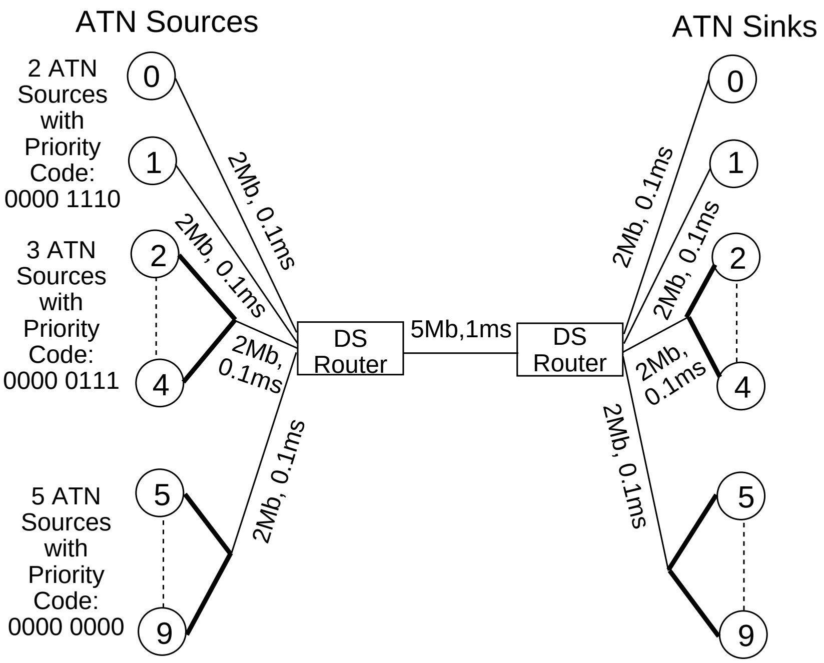

ATN Sources
ATN Sinks
2 ATN Sources with Priority Code: 0000 1110
3 ATN Sources with Priority Code: 0000 0111
5 ATN Sources with Priority Code: 0000 0000
0
1
2
4
5
9
2Mb, 0.1ms
DS Router
5Mb,1ms
DS Router

Figure 6: Network configuration

5.1 Performance Criteria

To show the effectiveness of our mapping strategy in providing QoS to end ATN applications, we have used goodput, queue size and drop ratio as the performance criteria. In the next section, we present the results of measurements of the above quantities from our simulation experiments.

5.2 Simulation Cases

We use the following four simulation cases to determine the QoS obtained by ATN sources.

- **Case 1: No congestion:** The traffic generated by the each high, medium and normal priority sources were set to 1 Mb, 0.666 Mb and 0.2 Mb respectively. According to the network configuration described in Section 4.2, there are two, three and five sources generating high, medium and normal priority traffic of 2Mb, 2Mb and 1Mb respectively. The amount of traffic of different priority are equal to the corresponding output link bandwidth assigned by scheduler described in Section 4.2. Under this scenario, *there should not be any significant congestion* at the edge DiffServ router because the sum of the traffic from the sources is equal to the bandwidth of the bottleneck link.

- **Case 2: Normal priority traffic gets into congestion:** The traffic generated by the each high, medium and normal priority sources were set to 1 Mb, 0.666 Mb and 0.6 Mb respectively. According to the network configuration described in Section 4.2, there are two, three and five sources generating high, medium and normal priority traffic of 2Mb, 2Mb and 3Mb respectively. The amount of traffic of high and medium priority are still equal to the corresponding output link bandwidth assigned by scheduler described in Section 4.2. However, the amount of traffic of normal priority is greater than its corresponding output link bandwidth. Under this scenario, the *normal priority traffic gets into congestion* at the edge Diffserv router.

- **Case 3: Medium priority traffic gets into congestion:** The traffic generated by the each high, medium and normal priority sources were set to 1Mb, 1.333 Mb and 0.2 Mb respectively. According to the network configuration described in Section 4.2, there are two, three and five sources generating high, medium and normal priority traffic of 2Mb, 4Mb and 1Mb respectively. The amount of traffic of high and normal priority are still equal to the corresponding output link bandwidth assigned by scheduler described in Section 4.2. However, the amount of traffic of medium priority is greater than its corresponding output link bandwidth. Under this scenario, the *medium priority traffic gets into congestion* at the edge Diffserv router.

- **Case 4: Both medium and normal priority traffics get into congestion:** The traffic generated by the each high, medium and normal priority sources were set to 1Mb, 1.333 Mb and 0.6 Mb respectively. According to the network configuration described in Section 4.2, there are two, three and five sources generating high, medium and normal priority traffic of 2Mb, 4Mb and 3Mb respectively. The amount of traffic of high priority is still equal to the corresponding output link bandwidth assigned by scheduler described in Section 4.2. However, the amount of traffic of both medium and normal priority are greater than their corresponding output link bandwidth. Under this scenario, *both medium and normal priority traffics get into congestion* at the edge Diffserv router.

5.3 Numerical Results

Table 2 shows the goodput of each ATN source for four different cases described in Section 5.2. Table 3 shows the drop ratio measured at the scheduler for four cases of the three different types of ATN sources. Figures 7, 8, 9 and 10 show the queue size for each of the four case (from *Case 1* to *Case 4*), from which the queuing delay and jitter can be evaluated.

Table 2: Goodput of each ATN source (Unit: Kb/S)

Sources		*Case 1*	*Case 2*	*Case 3*	*Case 4*
High priority Sources	Source 0	999.9990	999.9990	999.9990	999.9990
	Source 1	999.9990	999.9990	999.9990	999.9990
Medium priority Sources	Source 2	666.6660	666.6660	668.2409	668.4719
	Source 3	666.6660	666.6660	667.3379	667.5270
	Source 4	666.6660	666.6660	664.4189	663.9990
Normal priority Sources	Source 5	200.0039	199.6469	200.0039	199.4790
	Source 6	200.0039	201.8520	200.0039	201.9780
	Source 7	200.0039	202.4190	200.0039	201.6840
	Source 8	199.9830	199.8779	199.9830	200.4660
	Source 9	200.0039	196.2030	200.0039	196.3920

Case 1 is an ideal case. Each type of source (high, medium and normal priority sources) generates traffic at the rate equal to the bandwidth assigned by the scheduler. Therefore, there is no significant network congestion at the edge Diffserv router. As seen in Table 2, the goodput of each source is almost the same as its traffic generation rate. From Table 3, the drop ratio of each

Table 3: Drop ratio of ATN traffic.

Type of traffic	*Case 1*	*Case 2*	*Case 3*	*Case 4*
High priority Traffic	0.000000	0.000000	0.000000	0.000000
Medium priority Traffic	0.000000	0.000000	0.499817	0.499834
Normal priority Traffic	0.000000	0.665638	0.000000	0.665616

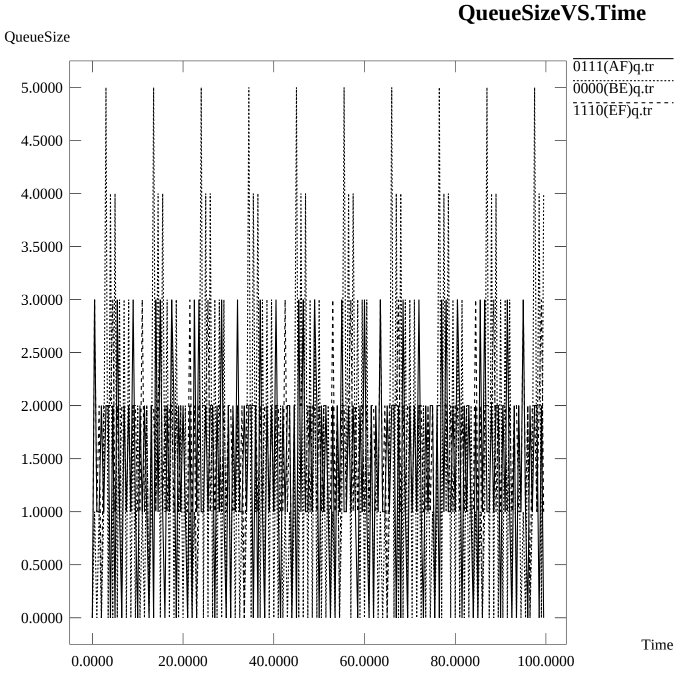


Figure 7: Queue size plots for *Case 1*

type of sources is zero. Figure 7 shows the queuing performance of each queue. Because this is an ideal case, the size of each queue is very small. Though the three queues have almost the same average size, we observe that the normal priority queue (mapping to BE queue, according to the mapping function) has the largest jitter. delay).

In **case 2**, we increased the traffic generation rate of normal priority sources, keeping the rates of the other two types of traffic unchanged. The traffic generating rate of each normal priority source is set to 0.6Mb. In this case, the normal priority traffic gets congested. As shown by Table 3, the drop ratio of normal priority traffic is greatly increased. However, drop ratio for the other two sources still remain at zero. As seen in Table 2, the goodput of normal priority traffic for each source is only about 0.2Mb, instead of the traffic generation rate of 0.6Mb. The reason is that the total available output bandwidth of normal priority traffic has been assigned to 1Mb by scheduler. From Figure 8, we find that the average queue size of the normal priority queue is far greater than the other two types of sources. In addition, the jitter of normal priority traffic is also greater than the other two types of sources. The high priority traffic has the smallest average queue size and the smallest jitter.

Case 3 is very similar to case 2. The only difference is that the medium priority traffic, rather than normal priority traffic, gets into congestion. As expected, we find the drop ratio of medium

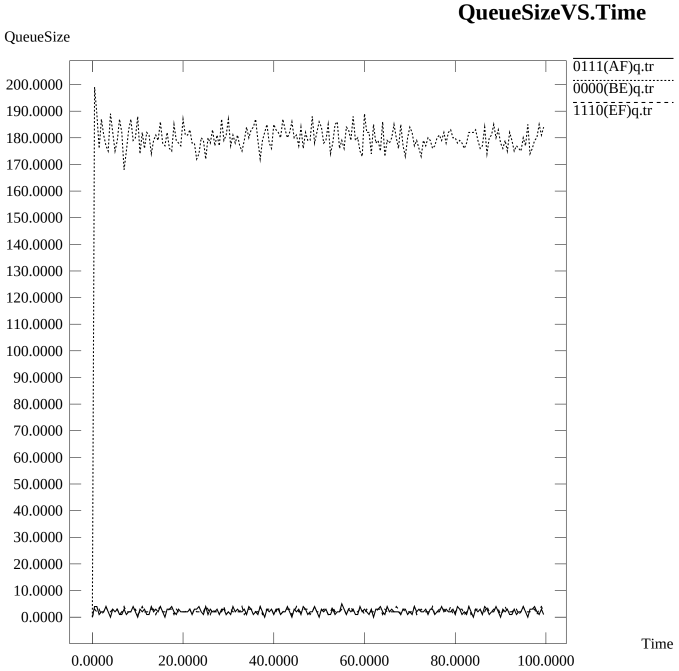


Figure 8: Queue size plots for *Case 2*.

priority traffic is increased with other two traffic types remaining at zero, and the goodput is also limited by the output link bandwidth assigned by the scheduler (which is 2Mb). From Figure 9, we find that both the jitter and the average queue size of medium priority traffic are far greater than the other two traffic types. The high priority traffic has the smallest average queue size and the smallest jitter.

In **Case 4**, we increased the traffic generation rates of both medium and normal priority sources. Both of them get into network congestion in this case. We find from Table 3 that the drop ratio of high priority traffic remains at zero, and drop ratios of both medium priority traffic and normal priority traffic are greatly increased. Furthermore, the drop ratio of normal priority traffic is greater than that of medium priority traffic. As shown by Table 2, the goodput of both the medium and normal priority traffic are limited by their link bandwidths allocated by scheduler. From Figure 10, we see that the normal priority traffic has both the biggest jitter and biggest average queue size. We can also find that the high priority traffic has both the smallest jitter and smallest average queue size.

From the above results, we can arrive at the following *observations*:

- The high priority traffic always has the smallest jitter, the smallest average queue size and the smallest drop ratio without being affected by the performance of other traffic. In other words, the high priority traffic receives the highest priority, which satisfies the priority requirements of ATN.

- The medium priority traffic has smaller drop ratio, jitter and queue size than the normal priority traffic, even in the presence of network congestion. This also satisfies the priority requirements of ATN.

We therefore, *conclude that the priority requirements of ATN can be successfully achieved when ATN traffic is mapped to the DiffServ domain in next generation Internet.*

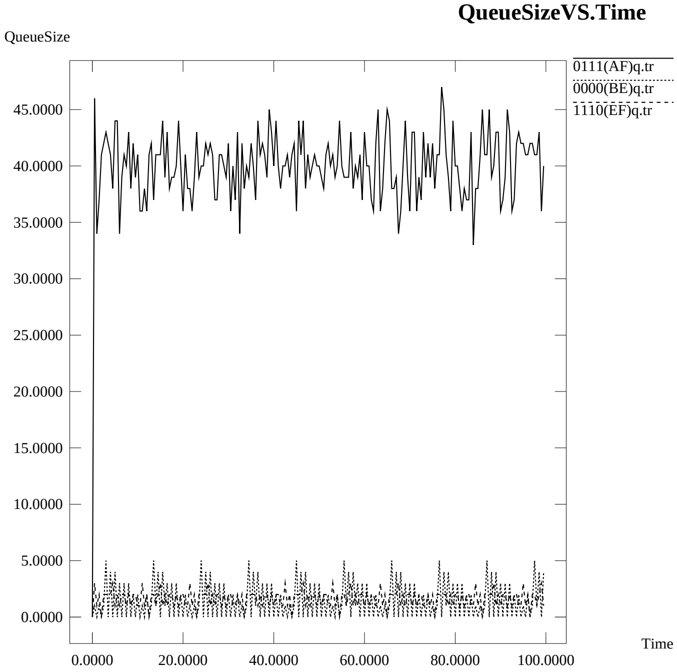


Figure 9: Queue size plots for *Case 3*

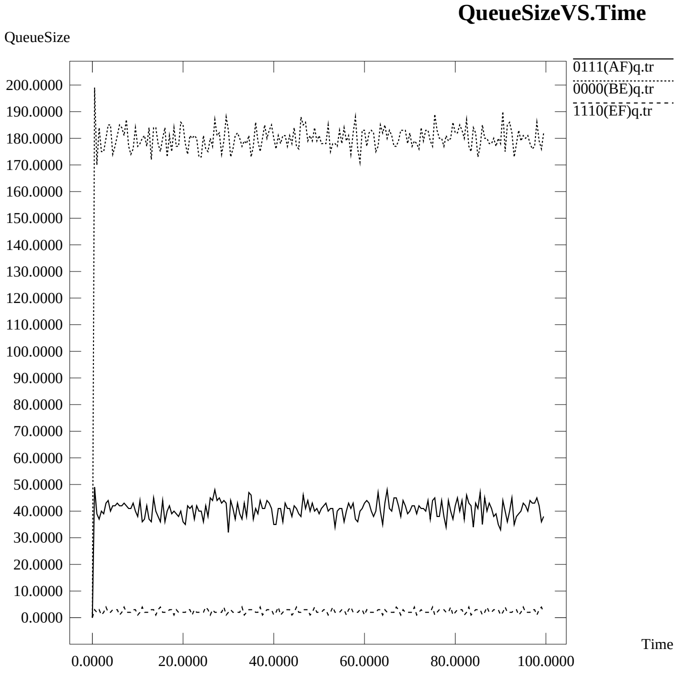


Figure 10: Queue size plots for *Case 4*

6 Conclusion

In this paper, we have proposed DiffServ as the backbone network to interconnect ATN subnetworks. We have designed a mapping function to map traffic flows coming from ATN with different priorities (indicated by the *priority* field in ATN packet header) to the corresponding PHBs in the DiffServ domain.

The proposed scheme has been studied in detail using simulation. It has been found that the QoS requirements of ATN can be achieved when ATN runs over DiffServ. We have illustrated our scheme by mapping ATN traffic of three different priorities to the three service classes of DiffServ. The ability of our scheme to provide QoS to end ATN applications has been demonstrated by measuring the drop ratio, goodput and queue size. We found that the high priority ATN traffic has the smallest jitter, the smallest average queue size and the smallest drop ratio, and is unaffected by the performance of other traffic. Moreover, the medium priority ATN traffic has a smaller drop ratio, jitter and queue size than the normal traffic, even in the presence of network congestion.

Acknowledgements

The authors thank NASA for support of this project. Faruque Ahamed carried out some initial work on this project.

References

[1] ATNP, "ATN SARPs - 2nd edition." Final Editor's drafts of the ATN SARPs, December 1999.

[2] R. Braden, D. Clark, and S. Shenker, "Integrated services in the internet architecture: an overview." RFC 1633, June 1994.

[3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services." RFC 2475, December 1998.

[4] ATNP, "Comprehensive ATN manual (CAMAL)." Final Editor's drafts of the ATN Guidance Material, January 1999.

[5] H. Hof, "Change proposal for improved text on the ATN priority architecture." ATNP/WG2/WP174, October 1995.

[6] K. Nichols, S.Blake, F. Baker, and D.Black, "Definition of the differentiated services field (DS field) in the IPv4 and IPv6 headers." RFC 2474, December 1998.

[7] Y. Bernet, S. Blake, D. Grossman, and A. Smith, "An informal management model for diffserv routers." draft-ietf-diffserv-model-04.txt, July 2000.

[8] V. Jacobson, K. Nichols, and K. Poduri, "An expedited forwarding PHB." RFC 2598, June 1999.

[9] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured forwarding PHB group." RFC 2597, June 1999.

[10] VINT Project U.C. Berkeley/LBNL, "ns v2.1b6: Network simulator." http://www-mash.cs.berkeley.edu/ns/, January 2000.

[11] S. Floyd and V. Jacobson, “Random early detection gateways for congestion avoidance,” *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, August 1993.

Paper 4

A. Durresi, S. Kota, M. Goyal, R. Jain, V. Bharani,
“Achieving QoS for TCP traffic in Satellite Networks
with Differentiated Services”

Achieving QoS for TCP traffic in Satellite Networks with Differentiated Services*

Arjan Durresi[1], Sastri Kota[2], Mukul Goyal[1], Raj Jain[3], Venkata Bharani[1]

[1]Department of Computer and Information Science, The Ohio State University,
2015 Neil Ave, Columbus, OH 43210-1277, USA
Tel: 614-688-5610, Fax: 614-292-2911, Email: {durresi, mukul, bharani}@cis.ohio-state.edu

[2]Lockheed Martin
60 East Tasman Avenue MS: C2135
San Jose CA 95134, USA
Tel: 408-456-6300/408- 473-5782, Email: sastri.kota@lmco.com

[3]Nayna Networks, Inc.
157 Topaz St., Milpitas, CA 95035, USA
Tel: 408-956-8000X309, Fax: 408-956-8730, Email: raj@nayna.com

ABSTRACT

Satellite networks play an indispensable role in providing global Internet access and electronic connectivity. To achieve such a global communications, provisioning of quality of service (QoS) within the advanced satellite systems is the main requirement. One of the key mechanisms of implementing the quality of service is traffic management. Traffic management becomes a crucial factor in the case of satellite network because of the limited availability of their resources. Currently, Internet Protocol (IP) only has minimal traffic management capabilities and provides best effort services. In this paper, we presented a broadband satellite network QoS model and simulated performance results. In particular, we discussed the TCP flow aggregates performance for their good behavior in the presence of competing UDP flow aggregates in the same assured forwarding. We identified several factors that affect the performance in the mixed environments and quantified their effects using a full factorial design of experiment methodology.

Correspondence address: Dr. Arjan Durresi,
Department of Computer and Information Science,
The Ohio State University, 2015 Neil Ave, Columbus, OH 43210, USA
Tel: 614-688-5610, Fax: 614-292-2911, Email: durresi@cis.ohio-state.edu

* This research was sponsored in part by grants from NASA Glenn Research Center, Cleveland, and OAI, Cleveland, Ohio

1. INTRODUCTION

The increasing worldwide demand for more bandwidth and Internet access is creating new opportunities for the deployment of global next generation satellite networks. Today it is clear that satellite networks will be a significant player in the digital revolution, and will specially benefit from on-board digital processing and switching, as well as other such technological advances as emerging digital compression, narrow spot beams for frequency reuse, digital intersatellite links, advanced link access methods and multicast technologies. Many new satellite communication systems have been planned and are under development including at Ka, Q/V-bands [7]. Some of the key design issues for satellite networks include efficient resource management schemes and QoS architectures.

However, satellite systems have several inherent constraints. The resources of the satellite communication network, especially the satellite and the Earth station, are expensive and typically have low redundancy; these must be robust and be used efficiently. The large delays in geostationary Earth orbit (GEO) systems and delay variations in low Earth orbit (LEO) systems affect both real-time and non-real-time applications. In an acknowledgement and time-out-based congestion control mechanism (like TCP), performance is inherently related to the delay-bandwidth product of the connection. Moreover, TCP round-trip time (RTT) measurements are sensitive to delay variations that may cause false timeouts and retransmissions. As a result, the congestion control issues for broadband satellite networks are somewhat different from those of low-latency terrestrial networks. Both interoperability issues as well as performance issues need to be addressed before a transport-layer protocol like TCP can satisfactorily work over long-latency satellite IP ATM networks.

There has been an increased interest in developing Differentiated Services (DS) architecture for provisioning IP QoS over satellite networks. DS aims to provide scalable service differentiation in the Internet that can be used to permit differentiated pricing of Internet service [1]. This differentiation may either be quantitative or relative. DS is scalable as traffic classification and conditioning is performed only at network boundary nodes. The service to be received by a traffic is marked as a code point in the DS field in the IPv4 or IPv6 header. The DS code point in the header of an IP packet is used to determine the Per-Hop Behavior (PHB), i.e. the forwarding treatment it will receive at a network node. Currently, formal specification is available for two PHBs - Assured Forwarding [4] and Expedited Forwarding [5]. In Expedited Forwarding, a transit node uses policing and shaping mechanisms to ensure that the maximum arrival rate of a traffic aggregate is less than its minimum departure rate. At each transit node, the minimum departure rate of a traffic aggregate should be configurable and independent of other traffic at the node. Such a per-hop behavior results in minimum delay and jitter and can be used to provide an end-to-end `Virtual Leased Line' type of service.

In Assured Forwarding (AF), IP packets are classified as belonging to one of four traffic classes. IP packets assigned to different traffic classes are forwarded independent of each other. Each traffic class is assigned a minimum configurable amount of resources (link bandwidth and buffer space). Resources not being currently used by another PHB or an AF traffic class can optionally be used by remaining classes. Within a traffic class, a packet is assigned one of three levels of drop precedence (green, yellow, red). In case of congestion, an AF-compliant DS node drops low precedence (red) packets in preference to higher precedence (green, yellow) packets.

In this paper, we describe a wide range of simulations, varying several factors to identify the significant ones influencing fair allocation of excess satellite network resources among congestion sensitive and insensitive flows. The factors that we studied in Section 2 include *a)* number of drop precedence required (one, two, or three), *b)* percentage of reserved (highest drop precedence) traffic, *c)* buffer management (Tail drop or Random Early Drop with different parameters), and *d)* traffic types (TCP aggregates, UDP aggregates). Section 3 describes the simulation configuration and parameters and experimental design techniques. Section 4 describes Analysis Of Variation (ANOVA) technique. Simulation results for TCP and UDP, for reserve rate utilization and fairness are also given. Section 5 summarizes the study's conclusions.

2. QOS FRAME WORK

The key factors that affect the satellite network performance are those relating to bandwidth management, buffer management, traffic types and their treatment, and network configuration. Band width management relates to the algorithms and parameters that affect service (PHB) given to a particular aggregate. In particular, the number of drop precedence (one, two, or three) and the level of reserved traffic were identified as the key factors in this analysis.

Buffer management relates to the method of selecting packets to be dropped when the buffers are full. Two commonly used methods are tail drop and random early drop (RED). Several variations of RED are possible in case of multiple drop precedence. These variations are described in Section 3.

Two traffic types that we considered are TCP and UDP aggregates. TCP and UDP were separated out because of their different response to packet losses. In particular, we were concerned that if excess TCP and excess UDP were both given the same treatment, TCP flows will reduce their rates on packet drops while UDP flows will not change and get the entire excess bandwidth. The analysis shows that this is in fact the case and that it is important to give a better treatment to excess TCP than excess UDP.

In this paper, we used a simple network configuration which was chosen in consultation with other researchers interested in assured forwarding. This is a simple configuration, which we believe, provides most insight in to the issues and on the other hand will be typical of a GEO satellite network.

We have addressed the following QoS issues in our simulation study:

- Three drop precedence (green, yellow, and red) help clearly distinguish between congestion sensitive and insensitive flows.
- The reserved bandwidth should not be overbooked, that is, the sum should be less than the bottleneck link capacity. If the network operates close to its capacity, three levels of drop precedence are redundant as there is not much excess bandwidth to be shared.
- The excess congestion sensitive (TCP) packets should be marked as yellow while the excess congestion insensitive (UDP) packets should be marked as red.
- The RED parameters have significant effect on the performance. The optimal setting of RED parameters is an area for further research.

2.1 Buffer Management Classifications

Buffer management techniques help identify which packets should be dropped when the queues exceed a certain threshold. It is possible to place packets in one queue or multiple queues depending upon their color or flow type. For the threshold, it is possible to keep a single threshold on packets in all queues or to keep multiple thresholds. Thus, the accounting (queues) could be single or multiple and the threshold could be single or multiple. These choices lead to four classes of buffer management techniques:

1. Single Accounting, Single Threshold (SAST)
2. Single Accounting, Multiple Threshold (SAMT)
3. Multiple Accounting, Single Threshold (MAST)
4. Multiple Accounting, Multiple Threshold (MAMT)

Random Early Discard (RED) is a well known and now commonly implemented packet drop policy. It has been shown that RED performs better and provides better fairness than the tail drop policy. In RED, the drop probability of a packet depends on the average queue length which is an exponential average of instantaneous queue length at the time of the packet's arrival [3]. The drop probability increases linearly from 0 to max_p as average queue length increases from min_th to max_th. With packets of multiple colors, one can calculate average queue length in many ways and have multiple sets of drop thresholds for packets of different colors. In general, with multiple colors, RED policy can be implemented as a variant of one of four general categories: SAST, SAMT, MAST, and MAMT.

Single Average Single Threshold RED has a single average queue length and same min_th and max_th thresholds for packets of all colors. Such a policy does not distinguish between packets of different colors and can also be called color blind RED. In Single Average Multiple Thresholds RED, average queue length is based on total number of packets in the queue irrespective of their color. However, packets of different colors have different drop thresholds. For example, if maximum queue size is 60 packets, the drop thresholds for green, yellow and red packets can be {40/60, 20/40, 0/10}. In these simulations, we used Single Average Multiple Thresholds RED.

In Multiple Average Single/Multiple Threshold RED, average queue length for packets of different colors is calculated differently. For example, average queue length for a color can be calculated using number of packets in the queue with same or better color [2]. In such a scheme, average queue length for green, yellow and red packets will be calculated using number of green, yellow + green, red + yellow + green packets in the queue respectively. Another possible scheme is where average queue length for a color is calculated using number of packets of that color in the queue [8]. In such a case, average queue length for green, yellow and red packets will be calculated using number of green, yellow and red packets in the queue respectively. Multiple Average Single Threshold RED will have same drop thresholds for packets of all colors whereas Multiple Average Multiple Threshold RED will have different drop thresholds for packets of different colors.

3. SIMULATION CONFIGURATION AND PARAMETERS

Figure 1 shows the network configuration for simulations. The configuration consists of customers 1 through 10 sending data over the link between Routers 1, 2 and using the same AF traffic class. Router 1 is located in a satellite ground station. Router 2 is located in a GEO satellite and Router 3 is located in destination ground station. Traffic is one-dimensional with only ACKs coming back from the other side. Customers 1 through 9 carry an aggregated traffic coming from 5 Reno TCP sources each. Customer 10 gets its traffic from a single UDP source sending data at a rate of 1.28 Mbps. Common configuration parameters are detailed in Tables 1 and 2. All TCP and UDP packets are marked green at the source before being 'recolored' by a traffic conditioner at the customer site. The traffic conditioner consists of two 'leaky' buckets (green and yellow) that mark packets according to their token generation rates (called reserved/green and yellow rate). In two-color simulations, yellow rate of all customers is set to zero. Thus, in two-color simulations, both UDP and TCP packets will be colored either green or red. In three-color simulations, customer 10 (the UDP customer) always has a yellow rate of 0. Thus, in three-color simulations, TCP packets coming from customers 1 through 9 can be colored green, yellow or red and UDP packets coming from customer 10 will be colored green or red. All the traffic coming to Router 1 passes through a Random Early Drop (RED) queue. The RED policy implemented at Router 1 can be classified as Single Average Multiple Threshold RED as explained in Section 3.

We have used NS simulator version 2.1b4a [9] for these simulations. The code has been modified to implement the traffic conditioner and multi-color RED (RED_n).

3.1 Experimental Design

In this study, we performed full factorial simulations involving many factors, which are listen in Tables 3 and 4 for two-color simulations and in Tables 5, 6 for three-color simulations:

- *Green Traffic Rates:* Green traffic rate is the token generation rate of green bucket in the traffic conditioner. We have experimented with green rates of 12.8, 25.6, 38.4 and 76.8 kbps per customer. These rates correspond to a total of 8.5%, 17.1%, 25.6% and 51.2% of network capacity (1.5 Mbps). In order to understand the effect of green traffic rate, we also conduct simulations with green rates of 102.4, 128, 153.6 and 179.2 kbps for two-color cases. These rates correspond to 68.3%, 85.3%, 102.4% and 119.5% of network capacity respectively. In last two cases, we have oversubscribed the available network bandwidth. The Green rates used and the simulations sets are shown in Tables 3 and 5 for two and three-color simulations respectively.
- *Green Bucket Size:* 1, 2, 4, 8, 16 and 32 packets of 576 bytes each, shown in Tables 4 and 6.
- *Yellow Traffic Rate* (only for three-color simulations, Table 6): Yellow traffic rate is the token generation rate of yellow bucket in the traffic conditioner. We have experimented with yellow rates of 12.8 and 128 kbps per customer. These rates correspond to 7.7% and 77% of total capacity (1.5 Mbps) respectively. We used a high yellow rate of 128 kbps so that all excess (out of green rate) TCP packets are colored yellow and thus can be distinguished from excess UDP packets that are colored red.
- *Yellow Bucket Size* (only for three-color simulations, Table 6): 1, 2, 4, 8, 16, 32 packets of 576 bytes each.

- *Maximum Drop Probability:* Maximum drop probability values used in the simulations are listed in Tables 4 and 6.
- *Drop Thresholds* for red colored packets: The network resources allocated to red colored packets and hence the fairness results depend on the drop thresholds for red packets. We experimented with different values of drop thresholds for red colored packets so as to achieve close to best fairness possible. Drop thresholds for green packets have been fixed at {40,60} for both two and three-color simulations. For three-color simulations, yellow packet drop thresholds are {20,40}. Drop threshols are listed in Tables 4 and 6.

In these simulations, size of all queues is 60 packets of 576 bytes each. The queue weight used to calculate RED average queue length is 0.002. For easy reference, we have given an identification number to each simulation (Tables 3 and 5). The simulation results are analyzed using ANOVA techniques [6] briefly described in Section 8.

3.2 Performance Metrics

Simulation results have been evaluated based on utilization of reserved rates by the customers and the fairness achieved in allocation of excess bandwidth among different customers.

Utilization of reserved rate by a customer is measured as the ratio of green throughput of the customer and the reserved rate. Green throughput of a customer is determined by the number of green colored packets received at the traffic destination(s). Since in these simulations, the drop

thresholds for green packets are kept very high in the RED queue at Router 1, chances of a green packet getting dropped are minimal and ideally green throughput of a customer should
equal its reserved rate.

The fairness in allocation of excess bandwidth among n customers sharing a link can be computed using the following formula [6]:

$$Fairness \quad Index = \frac{\left(\sum x_i\right)^2}{n \times \sum\left(x_i^2\right)}$$

Where x_i is the excess throughput of the ith customer. Excess throughput of a customer is determined by the number of yellow and red packets received at the traffic destination(s).

4. SIMULATION RESULTS

Simulation results of two and three-color simulations are shown in Figures 2 and 3, where a simulation is identified by its Simulation ID listed in Tables 3 and 5. Figures 2a and 2b show the fairness achieved in allocation of excess bandwidth among ten customers for each of the two and three-color simulations respectively. It is clear from figure 2a that fairness is not good in two-color simulations. With three colors, there is a wide variation in fairness results with best results being close to 1. Fairness is zero in some of the two-color simulations. In these simulations, total reserved traffic uses all the bandwidth and there is no excess bandwidth available to share. Also, there is a wide variation in reserved rate utilization by customers in two and three-color simulations.

Figure 3 shows the reserved rate utilization by TCP and UDP customers. For TCP customers shown in Figures 3a and 3c, we have plotted the average reserved rate utilization in each simulation. In some cases, reserved rate utilization is slightly more than one. This is because token buckets are initially full which results in all packets getting green color in the beginning. Figures 3b and 3d show that UDP customers have good reserved rate utilization in almost all cases. In contrast, TCP customers show a wide variation in reserved rate utilization.

In order to determine the influence of different simulation factors on the reserved rate utilization and fairness achieved in excess bandwidth distribution, we analyze simulation results statistically using Analysis of Variation (ANOVA) technique. Section 4.1 gives a brief introduction to ANOVA technique used in the analysis. In later sections, we present the results of statistical analysis of two and three-color simulations, in Sections 4.2 and 4.3.

4.1 Analysis Of Variation (ANOVA) Technique

The results of a simulation are affected by the values (or levels) of simulation factors (e.g. green rate) and the interactions between levels of different factors (e.g. green rate and green bucket size). The simulation factors and their levels used in this simulation study are listed in Tables 3, 4, 5 and 6. Analysis of Variation of simulation results is a statistical technique used to quantify these effects. In this section, we present a brief account of Analysis of Variation technique. More details can be found in [6].

Analysis of Variation involves calculating the Total Variation in simulation results around the Overall Mean and doing Allocation of Variation to contributing factors and their interactions. Following steps describe the calculations:

1. Calculate the *Overall Mean* of all the values.

2. Calculate the individual effect of each level *a* of factor *A*, called the *Main Effect* of *a*:

Main Effect_a = Mean_a - Overall Mean

where, Main Effect_a is the main effect of level *a* of factor *A*, Mean_a is the mean of all results with *a* as the value for factor *A*.

The main effects are calculated for each level of each factor.

3. Calculate the *First Order Interaction* between levels *a* and *b* of two factors *A* and *B* respectively for all such pairs:

$\text{Interaction}_{a,b}$ = $\text{Mean}_{a,b}$ - (Overall Mean + Main Effect_a + Main Effect_b)

where, $\text{Interaction}_{a,b}$ is the interaction between levels *a* and *b* of factors *A* and *B* respectively, $\text{Mean}_{a,b}$ is mean of all results with *a* and *b* as values for factors *A* and *B*, Main Effect_a and Main Effect_b are main effects of levels *a* and *b* respectively.

4. Calculate the *Total Variation* as shown below:

Total Variation = $\sum(\text{result}^2)$ - (Num_Sims) × (Overall Mean^2)

where, $\sum(\text{result}^2)$ is the sum of squares of all individual results and Num_Sims is total number of simulations.

5. The next step is the *Allocation of Variation* to individual main effects and first order interactions. To calculate the variation caused by a factor *A*, we take the sum of squares of the main effects of all levels of *A* and multiply this sum with the number of experiments conducted with each level of *A*. To calculate the variation caused by first order interaction between two factors *A* and *B*, we take the sum of squares of all the first-order interactions between levels of *A* and *B* and multiply this sum with the number of experiments conducted with each combination of levels of *A* and *B*. We calculate the allocation of variation for each factor and first order interaction between every pair of factors.

4.2 ANOVA Analysis for Reserved Rate Utilization

Table 7 shows the Allocation of Variation to contributing factors for reserved rate utilization. As shown in Figures 3b and 3d, reserved rate utilization of UDP customers is almost always good for both two and three-color simulations. However, in spite of very low probability of a green packet getting dropped in the network, TCP customers are not able to fully utilize their reserved rate in all cases. The little variation in reserved rate utilization for UDP customers is explained largely by bucket size. Large bucket size means that more packets will get green color in the beginning of the simulation when green bucket is full. Green rate and interaction between green rate and bucket size explain a substantial part of the variation. This is because the definition of rate utilization metric has reserved rate in denominator. Thus, the part of the utilization coming from initially full bucket gets more weight for low reserved rate than for high reserved rates. Also, in two-color simulations for reserved rates 153.6 kbps and 179.2 kbps, the network is oversubscribed and hence in some cases UDP customer has a reserved rate utilization lower than one. For TCP customers, green bucket size is the main factor in determining reserved rate utilization. TCP traffic, because of its bursty nature, is not able to fully utilize its reserved rate unless bucket size is sufficiently high. In our simulations, UDP customer sends data at a uniform rate of 1.28 Mbps and hence is able to fully utilize its reserved rate even when bucket size is low. However, TCP customers can have very poor utilization of reserved rate if bucket size is not sufficient. The minimum size of the leaky bucket required to fully utilize the token generation rate depends on the burstiness of the traffic.

4.3 ANOVA Analysis for Fairness

Fairness results shown in Figure 2a indicate that fairness in allocation of excess network bandwidth is very poor in two-color simulations. With two colors, excess traffic of TCP as well as UDP customers is marked red and hence is given same treatment in the network. Congestion sensitive TCP flows reduce their data rate in response to congestion created by UDP flow. However, UDP flow keeps on sending data at the same rate as before. Thus, UDP flow gets most of the excess bandwidth and the fairness is poor. In three-color simulations, fairness results vary widely with fairness being good in many cases. Table 8 shows the important factors influencing fairness in three-color simulations as determined by ANOVA analysis. Yellow rate is the most important factor in determining fairness in three-color simulations. With three colors, excess TCP traffic can be colored yellow and thus distinguished from excess UDP traffic, which is colored red. Network can protect congestion sensitive TCP traffic from congestion insensitive UDP traffic by giving better treatment to yellow packets than to red packets. Treatment given to yellow and red packets in the RED queues depends on RED parameters (drop thresholds and max drop probability values) for yellow and red packets. Fairness can be achieved by coloring excess TCP packets as yellow and setting the RED parameter values for packets of different colors correctly. In these simulations, we experiment with yellow rates of 12.8 kbps and 128 kbps. With a yellow rate of 12.8 kbps, only a fraction of excess TCP packets can be colored yellow at the traffic conditioner and thus resulting fairness in excess bandwidth distribution is not good. However with a yellow rate of 128 kbps, all excess TCP packets are colored yellow and good fairness is achieved with correct setting of RED parameters. Yellow bucket size also explains a substantial portion of variation in fairness results for three-color simulations. This is because bursty TCP traffic can fully utilize its yellow rate only if yellow bucket size is sufficiently high. The interaction between yellow rate and yellow bucket size for three-color fairness results is because of the fact that minimum size of the yellow bucket required for fully utilizing the yellow rate increases with yellow rate.

It is evident that three colors are required to enable TCP flows get a fair share of excess network resources. Excess TCP and UDP packets should be colored differently and network should treat them in such a manner so as to achieve fairness. Also, size of token buckets should be sufficiently high so that bursty TCP traffic can fully utilize the token generation rates.

5. CONCLUSIONS

One of the goals of deploying multiple drop precedence levels in an Assured Forwarding traffic class on a satellite network is to ensure that all customers achieve their reserved rate and a fair share of excess bandwidth. In this paper, we analyzed the impact of various factors affecting the performance of assured forwarding. The key conclusions are:

- The key performance parameter is the level of green (reserved) traffic. The combined reserved rate for all customers should be less than the network capacity. Network should be configured in such a manner so that in-profile traffic (colored green) does not suffer any packet loss and is successfully delivered to the destination.

- If the reserved traffic is overbooked, so that there is little excess capacity, two drop precedence give the same performance as three.

- The fair allocation of excess network bandwidth can be achieved only by giving different treatment to out-of-profile traffic of congestion sensitive and insensitive flows. The reason is that congestion sensitive flows reduce their data rate on detecting congestion however congestion insensitive flows keep on sending data as before. Thus, in order to prevent congestion insensitive flows from taking advantage of reduced data rate of congestion sensitive flows in case of congestion, excess congestion insensitive traffic should get much harsher treatment from the network than excess congestion sensitive traffic. Hence, it is important that excess congestion sensitive and insensitive traffic is colored differently so that network can distinguish between them. Clearly, three colors or levels of drop precedence are required for this purpose.

- Classifiers have to distinguish between TCP and UDP packets in order to meaningfully utilize the three drop precedence.

- RED parameters and implementations have significant impact on the performance. Further work is required for recommendations on proper setting of RED parameters.

6. REFERENCES

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, An Architecture for Differentiated Services, RFC 2475, December 1998.

[2] D. Clark, W. Fang, Explicit Allocation of Best Effort Packet Delivery Service, IEEE/ACM Transactions on Networking, August 1998: Vol. 6, No.4, pp. 349-361.

[3] S. Floyd, V. Jacobson, Random Early Detection Gateways for Congestion Avoidance, IEEE/ACM Transactions on Networking, 1(4): 397-413, August 1993.

[4] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, Assured Forwarding PHB Group, RFC 2597, June 1999. [3] V. Jacobson, K. Nichols, K. Poduri, An Expedited Forwarding PHB, RFC 2598, June 1999.

[5] V. Jacobson, K. Nichols, K. Poduri, An Expedited Forwarding PHB, RFC 2598, June 1999.

[6] R. Jain, The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Simulation and Modeling, New York, John Wiley and Sons Inc., 1991.

[7] S. Kota, “Multimedia Satellite Networks: Issues and Challenges,” Proc. SPIE International Symposium on Voice, Video, and Data Communications, Boston, Nov 1-5, 1998.

[8] N. Seddigh, B. Nandy, P. Pieda, Study of TCP and UDP Interactions for the AF PHB, Internet Draft - Work in Progress, draft-nsbnpp-diffserv-tcpudpaf-00.pdf, June 1999.

[9] NS Simulator, Version 2.1, Available from http://www-mash.cs.berkeley.edu/ns.

Table 1: General Configuration Parameters used in Simulation

Simulation Time	100 seconds
TCP Window	64 packets
IP Packet Size	576 bytes
UDP Rate	1.28Mbps
Maximum queue size (for all queues)	60 packets

Table 2: Link Parameters used in Simulations

Link between UDP/TCPs and Customers:	
Link Bandwidth	10 Mbps
One way Delay	1 microsecond
Drop Policy	DropTail
Link between Customers (Sinks) and Router 1 (Router 3):	
Link Bandwidth	1.5 Mbps
One way Delay	5 microseconds
Drop Policy	DropTail
Link between Router 1 and Router 2:	
Link Bandwidth	1.5 Mbps
One way Delay	125 milliseconds
Drop Policy From Router 1 to Router 2	RED_n
Drop Policy From Router 2 to Router 1	DropTail
Link between Router 2 and Router 3:	
Link Bandwidth	1.5 Mbps
One way Delay	125 milliseconds
Drop Policy	DropTail

Table 3: Two-color Simulation Sets and their Green Rate

Simulation ID	Green Rate [kbps]
1-144	12.8
201-344	25.6
401-544	38.4
601-744	76.8
801-944	102.4
1001-1144	128
1201-1344	153.6
1401-1544	179.2

Table 4: Parameters, which combinations are used in each Set of two-color Simulations

Max Drop Drop Probability {Green, Red}	{0.1, 0.1}, {0.1, 0.5}, {0.1, 1}, {0.5, 1}, {0.5, 1}, {1, 1}
Drop Thresholds {Green, Red}	{40/60, 0/10}, {40/60, 0/20}, {40/60, 0/5}, {40/60, 20/40}
Green Bucket (in Packets)	1, 2, 4, 8, 16, 32

Table 5: Three-color Simulation Sets and their Green Rate

Simulation ID	Green Rate [kbps]
1-720	12.8
1001-1720	25.6
2001-2720	38.4
3001-3720	76.8

Table 6: Parameters, which combinations are used in each Set of three-color Simulations

Max Drop Drop Probability {Green, Yellow, Red}	{0.1, 0.5, 1}, {0.1, 1, 1}, {0.5, 0.5, 1}, {0.5, 1, 1}, {1, 1, 1}
Drop Thresholds {Green, Yellow, Red}	{40/60, 20/40, 0/10}, {40/60, 20/40, 0/20}
Yellow Rate [kbps]	12.8, 128
Green bucket Size (in packets)	1, 2, 4, 8, 16, 32
Yellow bucket Size (in packets)	1, 2, 4, 8, 16, 32

Table 7: Main Factors Influencing Reserved Rate Utilization Results

Factor/Interaction	Allocation of Variation (in %age)			
	In two-color Simulations		In three-color Simulations	
	TCP	UDP	TCP	UDP
Green Rate	8.86%	31.55%	2.21%	20.41%
Green Bucket Size	86.22%	42.29%	95.25%	62.45%
Green Rate - Green Bucket Size	4.45%	25.35%	1.96%	17.11%

Table 8: Main Factors Influencing Fairness Results in three-color Simulations

Factor/Interaction	Allocation of Variation (in %age)
Yellow Rate	41.36
Yellow Bucket Size	28.96
Interaction between Yellow Rate and Yellow Bucket Size	26.49

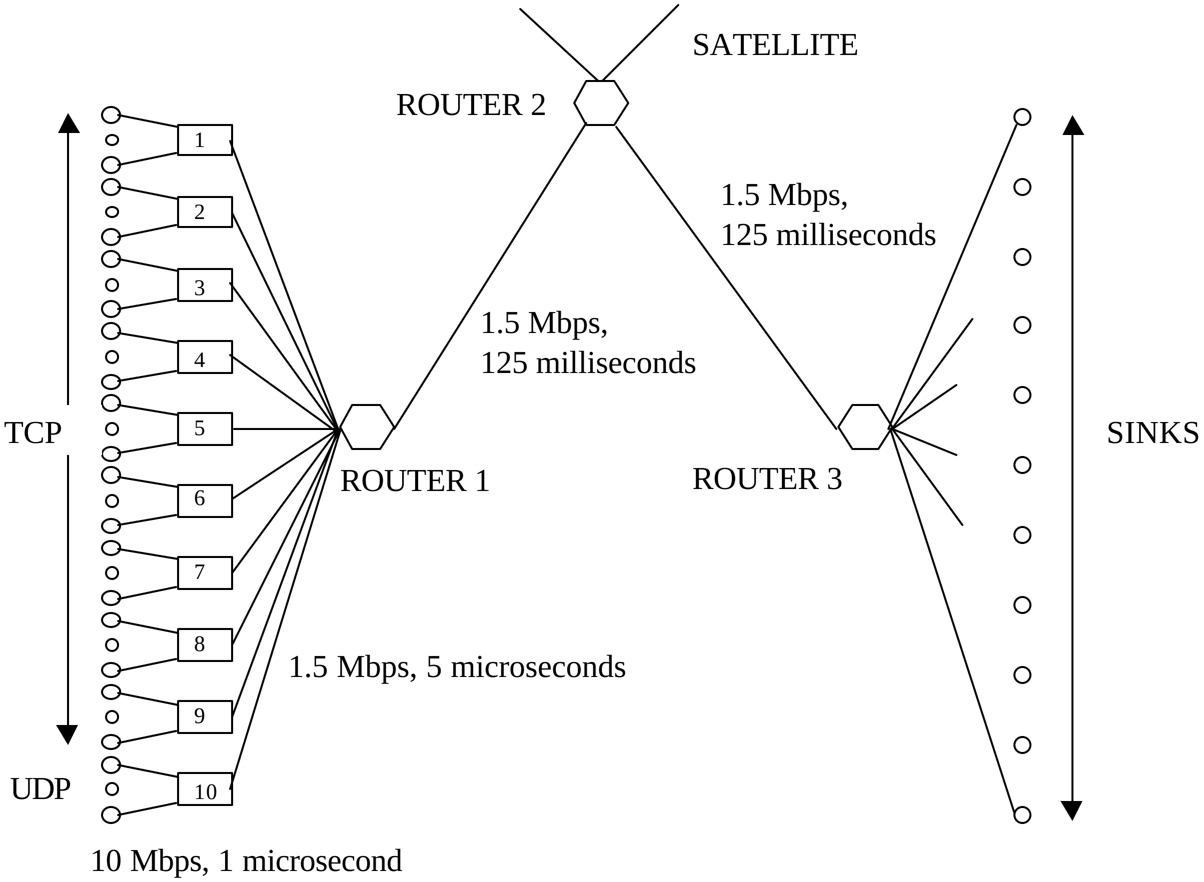


Figure 1. Simulation Configuration

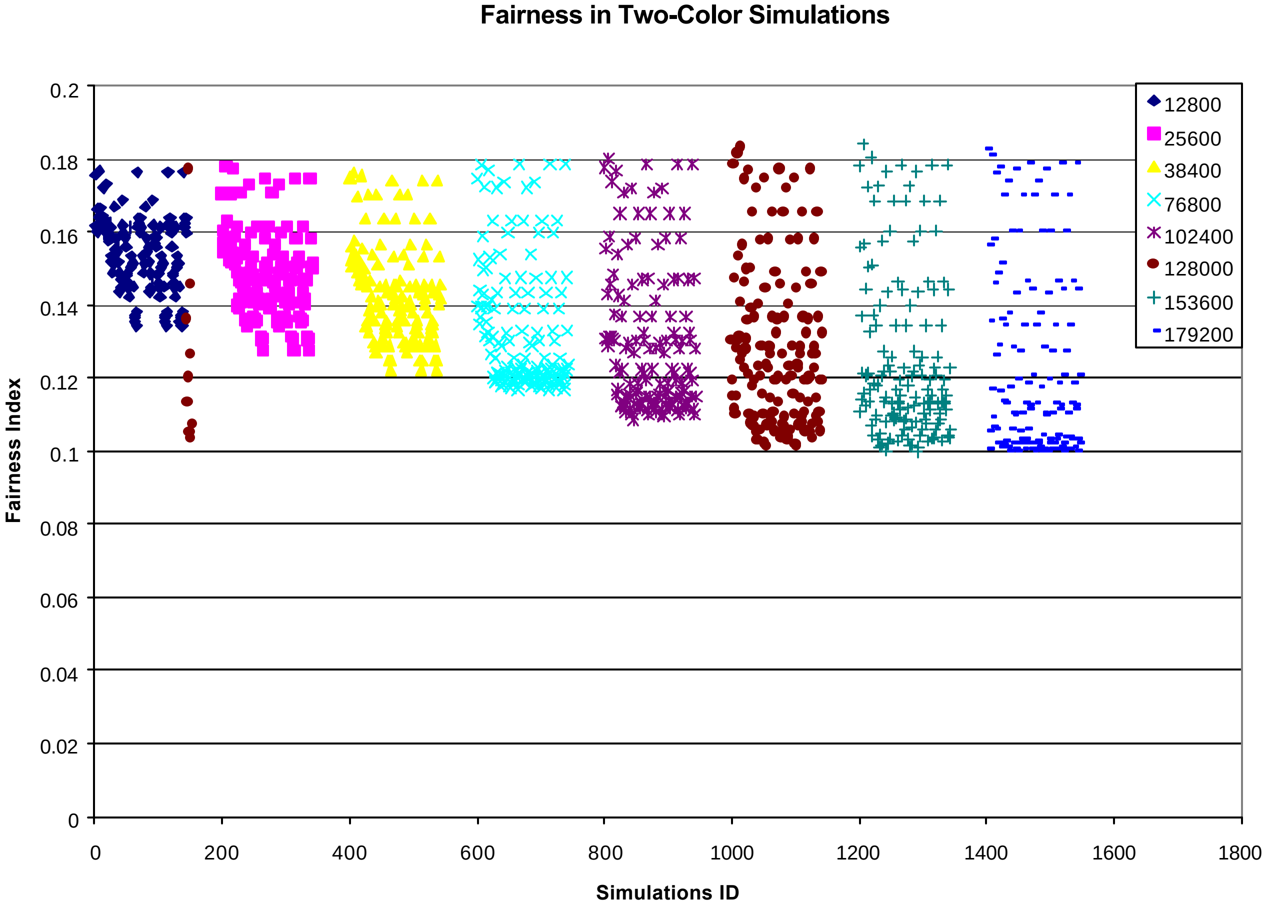


Figure 2a. Simulation Results: Fairness achieved in two-color Simulations with Different Reserved Rates

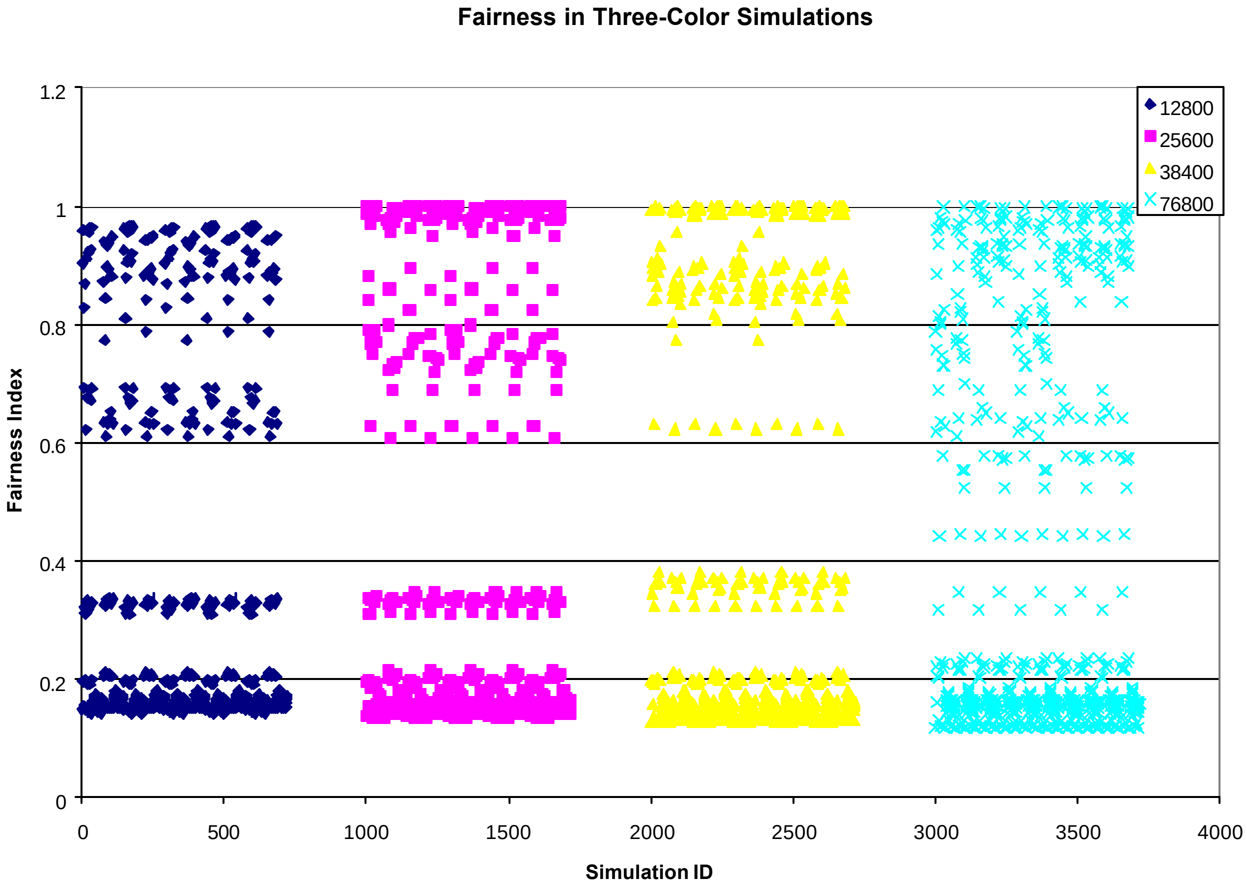


Figure 2b. Simulation Results: Fairness achieved in Three-Color Simulations with different Reserved Rates

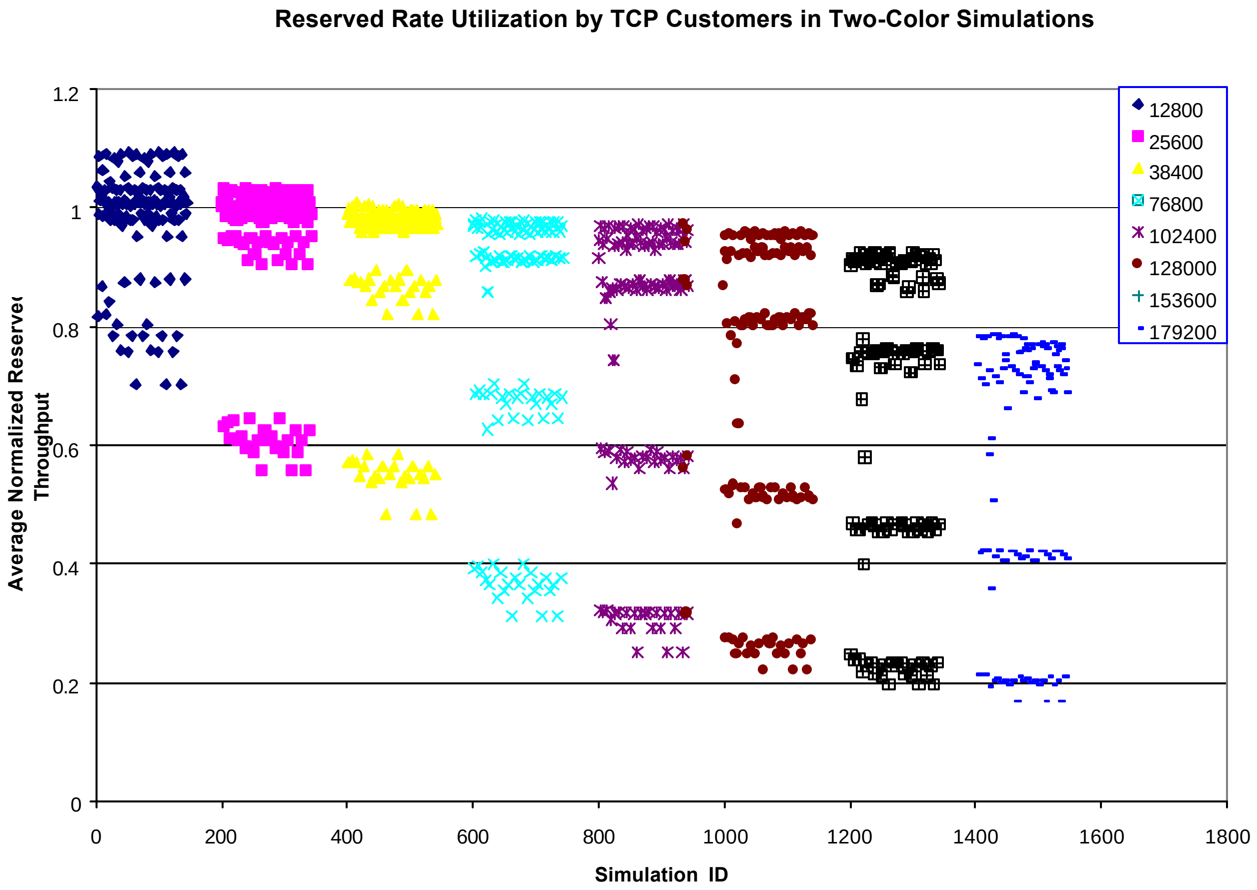


Figure 3a. Reserved Rate Utilization by TCP Customers in two-color Simulations

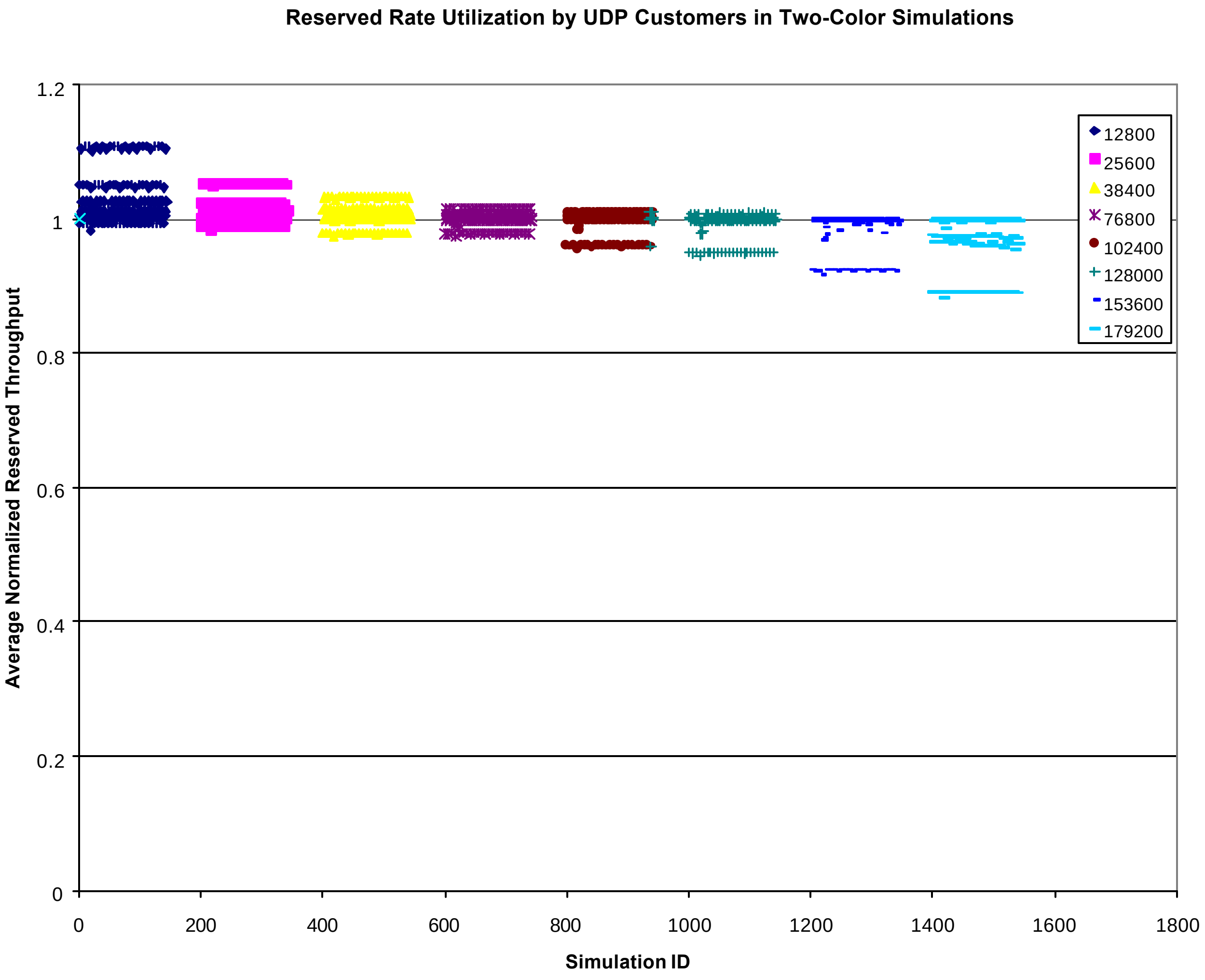


Figure 3b. Reserved Rate Utilization by UDP Customers in two-color Simulations

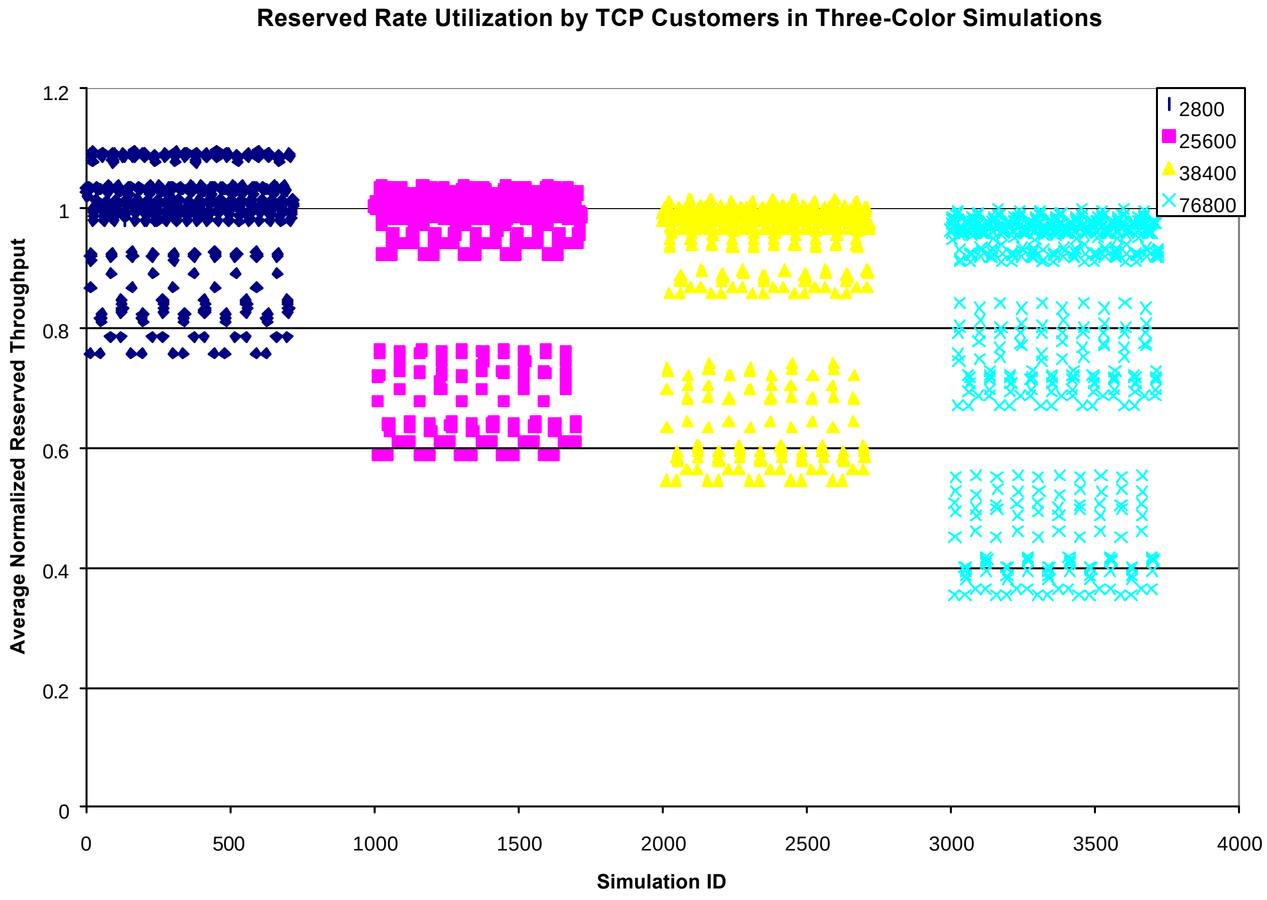


Figure 3c. Reserved Rate Utilization by TCP Customers in three-color Simulations

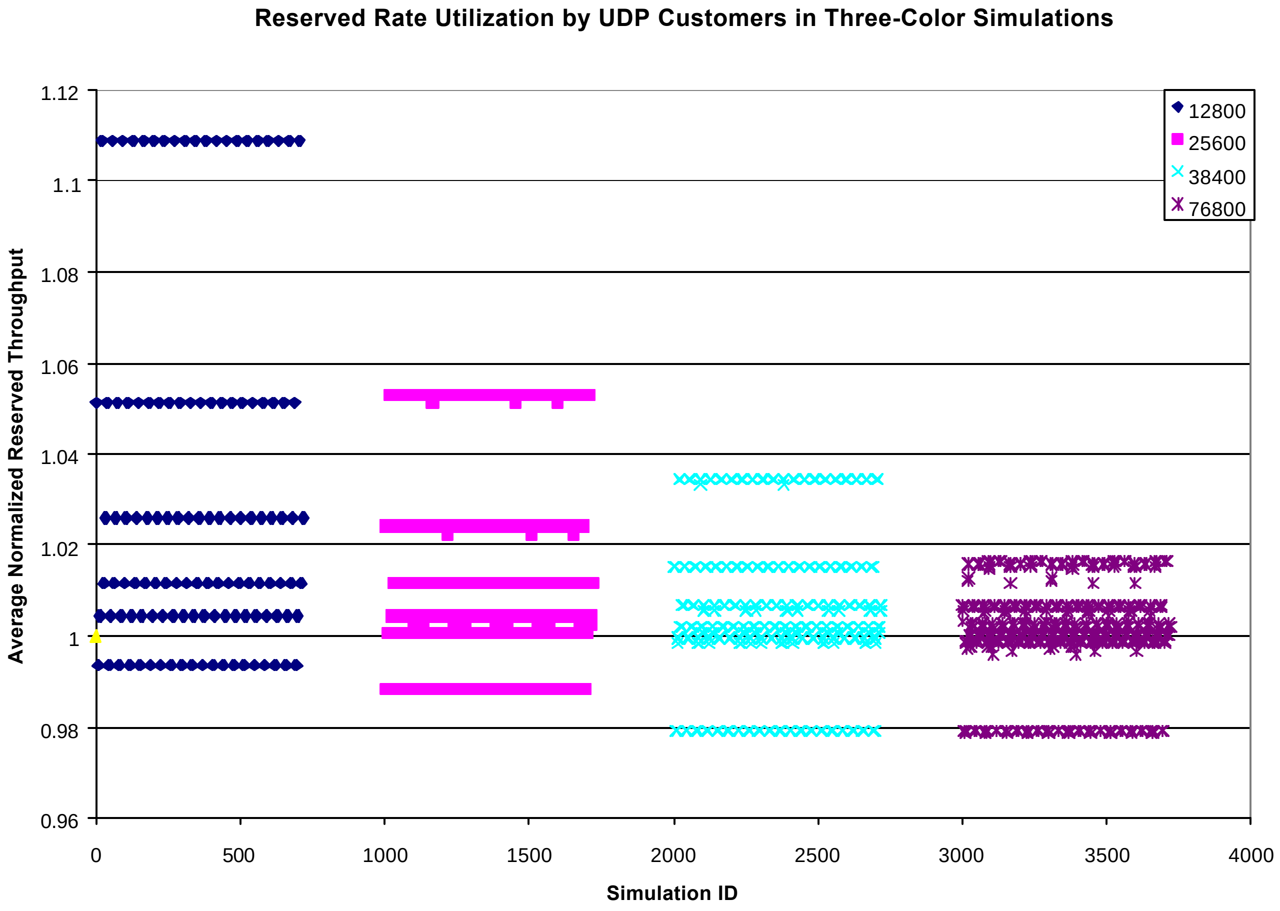


Figure 3d. Reserved Rate Utilization by UTP Customers in three-color Simulations

Paper 5

C. Liu and R. Jain, “Improving Explicit Congestion Notification with the Mark-Front Strategy”

Improving Explicit Congestion Notification with the Mark-Front Strategy [*]

Chunlei Liu
Department of Computer and Information Science
The Ohio State University, Columbus, OH 43210-1277
cliu@cis.ohio-state.edu

Raj Jain
Chief Technology Officer, Nayna Networks, Inc.
157 Topaz St, Milpitas, CA 95035
jain@acm.org

Abstract

Delivering congestion signals is essential to the performance of networks. Current TCP/IP networks use packet losses to signal congestion. Packet losses not only reduces TCP performance, but also adds large delay. Explicit Congestion Notification (ECN) delivers a faster indication of congestion and has better performance. However, current ECN implementations mark the packet from the tail of the queue. In this paper, we propose the mark-front strategy to send an even faster congestion signal. We show that mark-front strategy reduces buffer size requirement, improves link efficiency and provides better fairness among users. Simulation results that verify our analysis are also presented.

Keywords: Explicit Congestion Notification, mark-front, congestion control, buffer size requirement, fairness.

1 Introduction

Delivering congestion signals is essential to the performance of computer networks. In TCP/IP, congestion signals from the network are used by the source to determine the load. When a packet is acknowledged, the source increases its window size. When a congestion signal is received, its window size is reduced [1, 2].

TCP/IP uses two methods to deliver congestion signals. The first method is timeout. When the source sends a packet, it starts a retransmission timer. If it does not receive an acknowledgment within a certain time, it assumes congestion has happened in the network and the packet has been lost. Timeout is the slowest congestion signal because of the source has to wait a long time for the retransmission timer to expire.

The second method is loss detection. In this method, the receiver sends a duplicate ACK immediately on reception of each out-of-sequence packet. The source interprets the reception of three duplicate acknowledgments as a congestion packet loss. Loss detection can avoid the long wait of timeout.

Both timeout and loss detection use packet losses as congestion signals. Packet losses not only increase the traffic in the network, but also add large transfer delay. The Explicit Congestion Notification (ECN) proposed in [3, 4] provides a light-weight mechanism for routers to send a direct indication of congestion to the source. It makes use of two experimental bits in the IP header and two experimental bits in the TCP header. When the average queue length exceeds a threshold, the incoming packet is marked as *congestion experienced* with a probability calculated from the average queue length. When the marked packet is received, the receiver marks the acknowledgment using an *ECN-Echo* bit in the TCP header to send congestion notification back to the source. Upon receiving the ECN-Echo, the source halves its congestion window to help alleviate the congestion.

Many authors have pointed out that marking provides more information about the congestion state than packet dropping [5, 6], and ECN has been proven to be a better way to deliver congestion signal and exhibits a better performance [4, 5, 7].

[*] This research was sponsored in part by grants from Nokia Corporation, Burlington, Massachusetts and NASA Glenn Research Center, Cleveland, Ohio.

In most ECN implementations, when congestion happens, the congested router marks the incoming packet that just entered the queue. When the buffer is full or when a packet needs to be dropped as in Random Early Detection (RED), some implementations, such as the *ns* simulator [8], have the "drop from front" option as suggested by Yin [9] and Lakshman [10]. A brief discussion of drop from front in RED can be found in [11]. However, for packet marking, these implementations still pick the incoming packet and not the front packet. We call this policy "mark-tail".

In this paper, we propose a simple marking mechanism — the "mark-front" strategy. This strategy marks a packet when the packet is going to leave the queue and the queue length is greater than the pre-determined threshold. The mark-front strategy is different from the current mark-tail policy in two ways. First, since the router marks the packet at the time when it is sent, and not at the time when the packet is received, a more up-to-date congestion signal is carried by the marked packet. Second, since the router marks the packet in the front of the queue and not the incoming packet, congestion signals do not undergo the queueing delay as the data packets. In this way, a faster congestion feedback is delivered to the source.

The implementation of this strategy is extremely simple. One only needs to move the marking action from the enqueue procedure to the dequeue procedure and choose the packet leaving the queue in stead of the packet entering the queue.

We justify the mark-front strategy by studying its benefits. We find that, by providing faster congestion signals, mark-front strategy reduces the buffer size requirement at the routers; it avoids packet losses and thus improves the link efficiency when the buffer size in routers is limited. Our simulations also show that mark-front strategy improves the fairness among old and new users, and alleviates TCP's discrimination against connections with large round trip time.

The mark-front strategy differs from the "drop from front" option in that when packets are dropped, only implicit congestion feedback can be inferred from timeout or duplicate ACKs; when packets are marked, explicit and faster congestion feedback is delivered to the source.

Gibbons and Kelly [6] suggested a number of mechanisms for packet marking, such as "marking all the packets in the queue at the time of a packet loss", "marking every packet leaving the queue from the time of a packet loss until the queue becomes empty", and "marking packets randomly as they leave the queue with a probability so that later packets will not be lost." Our mark-front strategy differs from these marking mechanisms in that it is a simple marking rule that faithfully reflects the up-to-date congestion status, while the mechanisms suggested by Gibbons and Kelly either do not reflect the correct congestion status, or need sophisticated probability calculation about which no sound algorithm is known.

It is worth mentioning that mark-front strategy is as effective in high speed networks as in low speed networks. Lakshman and Madhow [12] showed that the amount of drop-tail switches should be at least two to three times the bandwidth-delay product of the network in order for TCP to achieve decent performance and to avoid losses in the slow start phase. Our analysis in section 4.3 reveals that in the steady-state congestion avoidance phase, the queue size fluctuates from empty to one bandwidth-delay product. So the queueing delay experienced by packets when congestion happens is comparable to the fixed round-trip time. [1] Therefore, the mark-front strategy can save as much as a fixed round-trip time in congestion signal delay, independent of the link speed.

We should also mention that the mark-front strategy applies to both wired and wireless networks. When the router threshold is properly set, the coherence between consecutive packets can be used to distinguish packet losses due to wireless transmission error from packet losses due to congestion. This result will be reported elsewhere.

This paper is organized as follows. In section 2 we describe the assumptions for our analysis. Dynamics of queue growth with TCP window control is studied in section 3. In section 4, we compare the buffer size requirements of mark-front and mark-tail strategies. In section 5, we explain why mark-front is fairer than mark-tail. The simulation results that verify our conclusions are presented in section 6. In section 7, we remove the assumptions made to facilitate the analysis, and apply the mark-front strategy to the RED algorithm. Simulation results show that mark-front has the advantages over mark-tail as revealed by the analysis.

[1] The fixed round-trip time is the round-trip time under light load, i.e., without queueing delay.

2 Assumptions

ECN is used together with TCP congestion control mechanisms like slow start and congestion avoidance [2]. When the acknowledgment is not marked, the source follows existing TCP algorithms to send data and increase the congestion window. Upon the receipt of an ECN-Echo, the source halves its congestion window and reduces the slow start threshold. In the case of a packet loss, the source follows the TCP algorithm to reduce the window and retransmit the lost packet.

ECN delivers congestion signals by setting the *congestion experienced* bit, but determining when to set the bit depends on the congestion detection policy. In [3], ECN is proposed to be used with average queue length and RED. Their goal is to avoid sending congestion signals caused by transient traffic and to desynchronize sender windows [13, 14]. In this paper, to allow analytical modeling, we assume a simplified congestion detection criterion: when the *actual queue length* is smaller than the threshold, the incoming packet will not be marked; when the *actual queue length* exceeds the threshold, the incoming packet will be marked.

We also make the following assumptions. (1) Receiver windows are large enough so the bottleneck is in the network. (2) Senders always have data to send and will send as many packets as their windows allow. (3) There is only one bottleneck link that causes queue buildup. (4) Receivers acknowledge every packet received and there are no delayed acknowledgments. (5) There is no ACK compression [15]. (6) The queue length is measured in packets and all packets have the same size.

3 Queue Dynamics with TCP Window Control

In this section, we study the relationship between the window size at the source and the queue size at the congested router. The purpose is to show the difference between mark-tail and mark-front strategies. Our analysis is made on one connection, but with small modifications, it can also apply to multiple connection case. Simulation results of multiple connections and connections with different round trip time will be presented in section 6.

In a path with one connection, the only bottleneck is the first link with the lowest rate in the entire route. In case of congestion, queue builds up only at the router before the bottleneck link. The following lemma is obvious.

Lemma 1 *If the data rate of the bottleneck link is d packets per second, then the downstream packet inter-arrival time and the ack inter-arrival time on the reverse link can not be shorter than $1/d$ seconds. If the bottleneck link is fully-loaded (i.e., no idling), then the downstream packet inter-arrival time and the ack inter-arrival time on the reverse link are $1/d$ seconds.*

Denote the source window size at time t as $w(t)$, then we have

Theorem 1 *Consider a path with only one connection and only one bottleneck link. Let the fixed round trip time be r seconds, the bottleneck link rate be d packets per second, and the propagation and transmission time between the source and bottleneck router be t_p. If the bottleneck link has been busy for at least r seconds, and a packet just arrived at the congested router at time t, then the queue length at the congested router is*

$$Q(t) = w(t - t_p) - rd. \tag{1}$$

Proof Consider the packet that just arrived at the congested router at time t. It was sent by the source at time $t - t_p$. At that time, the number of packets on the path and outstanding acks on the reverse link was $w(t - t_p)$. By time t, $t_p d$ acks are received by the source. All packets between the source and the router have entered the congested router or have been sent downstream. As shown in Figure 1, the pipe length from the congested router to the receiver, and then back to the source is $r - t_p$. The number of downstream packets and outstanding acks are $(r - t_p)d$. The rest of the $w(t - t_p)$ unacknowledged packets are still in the congested router. So the queue length is

$$Q(t) = w(t - t_p) - t_p d - (r - t_p)d = w(t - t_p) - rd. \tag{2}$$

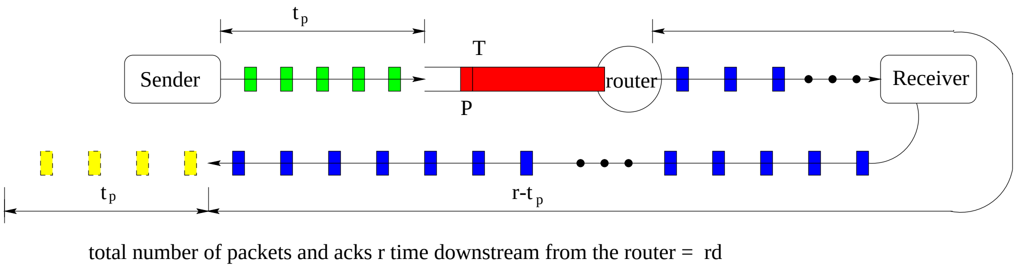

t p
T
Sender
router
Receiver
P
t p
r-t p
total number of packets and acks r time downstream from the router = rd

Figure 1: Calculation of the queue length

This finishes the proof.

Notice that in this theorem, we did not use the number of packets between the source and the congested router to estimate the queue length, because the packets downstream from the congested router and the acks on the reverse link are equally spaced, but the packets between the source and the congested router may not be.

The analysis in this theorem is based on the assumptions in section 2. The conclusion applies to both slow start and congestion avoidance phases. In order for equation (1) to hold, the router must have been congested for at least r seconds.

4 Buffer Size Requirement and Threshold Setting

When ECN signals are used for congestion control, the network can achieve zero packet loss. When acknowledgments are not marked, the source gradually increase the window size. Upon the receipt of an ECN-Echo, the source halves its congestion window to reduce the congestion.

In this section, we analyze the buffer size requirement for both mark-tail and mark-front strategies. The result also includes an analysis on how to set the threshold.

4.1 Mark-Tail Strategy

Suppose P was the packet that increased the queue length over the threshold T, and it was sent from the source at time s_0 and arrived at the congested router at time t_0. Its acknowledgment, which was an ECN-echo, arrived at the source at time s_1 and the window was reduced at the same time. We also assume that the last packet before the window reduction was sent at time s_1^- and arrived at the congested router at time t_1^-.

In order to use Theorem 1, we need to consider two cases separately: when T is large and when T is small, compared to rd.

Case 1 If T is reasonably large (about rd) such that the buildup of a queue of size T needs r time, the assumption in Theorem 1 is satisfied, we have

$$T = Q(t_0) = w(t_0 - t_p) - rd = w(s_0) - rd, \tag{3}$$

so

$$w(s_0) = T + rd. \tag{4}$$

Since the time elapse between s_0 and s_1 is one RTT, if packet P were not marked, the congestion window would increase to $2w(s_0)$. Since P was marked, the congestion window before receiving the ECN-Echo was

$$w(s_1^-) = 2w(s_0) - 1 = 2(T + rd) - 1. \tag{5}$$

When the last packet sent under this window reached the router at time t_1^-, the queue length was

$$Q(t_1^-) = w(s_1^-) - rd = 2w(s_0) - 1 - rd = 2T + rd - 1. \tag{6}$$

Upon the receipt of ECN-Echo, the congestion window was halved. The source can not send any more packets before half of the packets are acknowledged. So $2T + rd - 1$ is the maximum queue length.

Case 2 If T is small, rd is an overestimate of the number of downstream packets and acks on the reverse link.

$$w(s_0) = T + \textit{number of downstream packets and acks} \ \leq T + rd. \tag{7}$$

Therefore,

$$Q(t_1^-) = w(s_1^-) - rd = (2w(s_0) - 1) - rd \leq 2(T + rd) - 1 - rd = 2T + rd - 1. \tag{8}$$

So, in both cases, $2T + rd - 1$ is an upper bound of queue length that can be reached in slow start phase.

Theorem 2 *In a TCP connection with ECN congestion control, if the fixed round trip time is r seconds, the bottleneck link rate is d packets per second, and the bottleneck router uses threshold T for congestion detection, then the maximum queue length can be reached in slow start phase is less than or equal to $2T + rd - 1$.*

As shown by equation (6), when T is large, the bound $2T + rd - 1$ can be reached with equality. When T is small, $2T + rd - 1$ is just an upper bound. Since the queue length in congestion avoidance phase is smaller, this bound is actually the buffer size requirement.

4.2 Mark-Front Strategy

Suppose P was the packet that increased the queue length over the threshold T, and it was sent from the source at time s_0 and arrived at the congested router at time t_0. The router marked the packet P' that stood in the front of the queue. The acknowledgment of P', which was an ECN-echo, arrived at the source at time s_1 and the window was reduced at the same time. We also suppose the last packet before the window reduction was sent at time s_1^- and arrived at the congested router at time t_1^-.

Consider two cases separately: when T is large and when T is small.

Case 1 If T is reasonably large (about rd) such that the buildup of a queue of size T needs r time, the assumption in Theorem 1 is satisfied. We have

$$T = Q(t_0) = w(t_0 - t_p) - rd = w(s_0) - rd, \tag{9}$$

so

$$w(s_0) = T + rd. \tag{10}$$

In slow start phase, the source increases the congestion window by one for every acknowledgment it receives. If the acknowledgment of P was received at the source without the congestion indication, the congestion window would be doubled to

$$2w(s_0) = 2(T + rd).$$

However, when the acknowledgment of P' arrived, $T - 1$ acknowledgments corresponding to packets prior to P were still on the way. So the window size at time s_1^- was

$$w(s_1^-) = 2w(s_0) - (T - 1) - 1 = T + 2rd. \tag{11}$$

When the last packet sent under this window reached the router at time t_1^-, the queue length was

$$Q(t_1^-) = w(s_1^-) - rd = T + 2rd - rd = T + rd. \tag{12}$$

Upon the receipt of ECN-Echo, congestion window is halved. The source can not send any more packets before half of the packets are acknowledged. So $T + rd$ is the maximum queue length.

Case 2 If T is small, rd is an overestimate of the number of downstream packets and acks on the reverse link.

$$w(s_0) = T + \textit{number of downstream packets and acks } \leq T + rd. \tag{13}$$

Therefore,

$$Q(t_1^-) = w(s_1^-) - rd = (2w(s_0) - T) - rd \leq 2(T + rd) - T - rd = T + rd. \tag{14}$$

So, in both cases, $T + rd$ is an upper bound of queue length that can be reached in the slow start phase.

Theorem 3 *In a TCP connection with ECN congestion control, if the fixed round trip time is r seconds, the bottleneck link rate is d packets per second, and the bottleneck router uses threshold T for congestion detection, then the maximum queue length that can be reached in slow start phase is less than or equal to $T + rd$.*

Again, when T is large, equation (12) shows the bound $T + rd$ is tight. Since the queue length in congestion avoidance phase is smaller, this bound is actually the buffer size requirement.

Theorem 2 and 3 estimate the buffer size requirement for zero-loss ECN congestion control.

4.3 Threshold Setting

In the congestion avoidance phase, congestion window increases roughly by one in every RTT. Assuming mark-tail strategy is used, using the same timing variables as in the previous subsections, we have

$$w(s_0) = Q(t_0) + rd = T + rd. \tag{15}$$

The congestion window increases roughly by one in an RTT,

$$w(s_1^-) = T + rd + 1. \tag{16}$$

When the last packet sent before the window reduction arrived at the router, it saw a queue length of $T + 1$:

$$Q(t_1^-) = w(s_1^-) - rd = T + 1. \tag{17}$$

Upon the receipt of the ECN-Echo, the window was halved:

$$w(s_1) = (T + rd + 1)/2. \tag{18}$$

The source may not be able to send packets immediately after s_1. After some packets were acknowledged, the halved window allowed new packets to be sent. The first packet sent under the new window saw a queue length of

$$Q(t_1) = w(s_1) - rd = (T + rd + 1)/2 - rd = (T - rd + 1)/2. \tag{19}$$

The congestion window was fixed for an RTT and then began to increase. So $Q(t_1)$ was the minimum queue length in a cycle.

In summary, in the congestion avoidance phase, the maximum queue length is $T + 1$ and the minimum queue length is $(T - rd + 1)/2$.

In order to avoid link idling, we should have $(T - rd + 1)/2 \geq 0$ or equivalently, $T \geq rd - 1$. On the other hand, if $minQ$ is always positive, the router keeps an unnecessarily large queue and all packets suffer a long queueing delay. Therefore, the best choice of threshold should satisfy

$$(T - rd + 1)/2 = 0, \tag{20}$$

or

$$T = rd - 1. \tag{21}$$

If mark-front strategy is used, the source's congestion window increases roughly by one in every RTT, but congestion feedback travels faster than the data packets. Hence

$$Q(s_1^-) = T + rd + \epsilon, \tag{22}$$

where ϵ is between 0 and 1, and depends on the location of the congested router. Therefore,

$$Q(t_1^-) = w(s_1^-) - rd = T + \epsilon, \tag{23}$$

$$w(s_1) = (T + rd + \epsilon)/2, \tag{24}$$

$$Q(t_1) = w(s_1) - rd = (T + rd + \epsilon)/2 - rd = (T - rd + \epsilon)/2. \tag{25}$$

For the reason stated above, the best choice of threshold is $T = rd - \epsilon$. Compared with rd, the difference between $rd - \epsilon$ and $rd - 1$ can be ignored. So we have the following theorem:

Theorem 4 *In a path with only one connection, the optimal threshold that achieves full link utilization while keeping queueing delay minimal in congestion avoidance phase is $rd - 1$. If the threshold is smaller than this value, the link will be under-utilized. If the threshold is greater than this value, the link can be full utilized, but packets will suffer an unnecessarily large queueing delay.*

Combining the results in Theorem 2, 3 and 4, we can see that the mark-front strategy reduces the buffer size requirement from about $3rd$ to $2rd$. It also reduces the congestion feedback's delay by one fixed round-trip time.

5 Lock-out Phenomenon and Fairness

One of the weaknesses of mark-tail policy is its discrimination against new flows. Consider the time when a new flow joins the network, but the buffer of the congested router is occupied by packets of old flows. In the mark-tail strategy, the packet that just arrived will be marked, but the packets already in the buffer will be sent without being marked. The acknowledgments of the sent packets will increase the window size of the old flows. Therefore, the old flows which already have large share of the resources will grow even larger. However, the new flow with small or no share of the resources has to back off, since its window size will be reduced by the marked packets. This causes a "lock-out" phenomenon in which a single connection or a few flows monopolize the buffer space and prevent other connections from getting room in the queue [16]. Lock-out leads to gross unfairness among users and is clearly undesirable.

Contrary to the mark-tail policy, the mark-front strategy marks the packets in the buffer first. Connections with large buffer occupancy will have more packets marked than connections with small buffer occupancy. Compared with the mark-tail strategy that let the packets in the buffer escape the marking, mark-front strategy helps to prevent the lock-out phenomenon. Therefore, we can expect that mark-front strategy to be fairer than mark-tail strategy.

TCP's discrimination against connections with large RTT is also well known. The cause of this discrimination is similar to the discrimination against new connections. If connections with small RTT and large RTT start at the same time, the connections with small RTT will receive their acknowledgment faster and therefore grow faster. When congestion happens, connections with small RTT will take more buffer room than connections with large RTT. With mark-tail policy, packets already in the queue will not be marked but only newly arrived packets will be marked. Therefore, connections with small RTT will grow even larger, but connections with large RTT have to back off. Mark-front alleviates this discrimination by treating all packets in the buffer equally. Packets already in the buffer may also be marked. Therefore, connections with large RTT can have larger bandwidth.

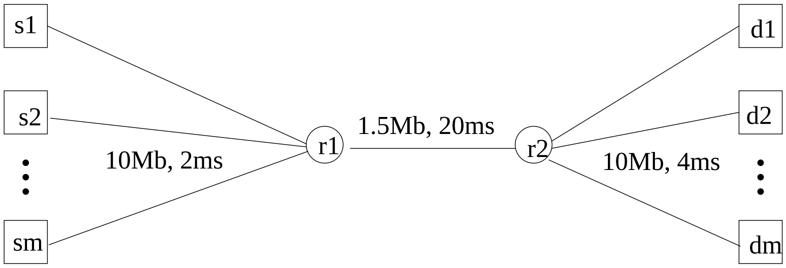


Figure 2: Simulation model.

6 Simulation Results

In order to compare the mark-front and mark-tail strategies, we performed a set of simulations with the ns simulator [8]. We modified the RED algorithm in ns simulator to deterministically mark the packets when the real queue length exceeds the threshold. The basic simulation model is shown in Figure 2. A number of sources $s_1, s_2, \ldots, s_m$ are connected to the router r_1 by 10 Mbps links, router r_1 is connected to r_2 by a 1.5 Mbps link, and destinations $d_1, d_2, \ldots, d_m$ are connected to r_2 by 10 Mbps links. The link speeds are chosen so that congestion will only happen at the router r_1, where mark-tail and mark-front strategies are tested.

With the basic configuration shown in Figure 2, the fixed round trip time, including the propagation time and the transmission time at the routers, is 59 ms. Changing the propagation delay between router r_1 and r_2 from 20 ms to 40 ms gives an RTT of 99 ms. Changing the propagation delays between the sources and router r_1 gives us configurations of different RTT. An FTP application runs on each source. Reno TCP and ECN are used for congestion control. The data packet size, including all headers, is 1000 bytes and the acknowledgment packet size is 40 bytes.

With the basic configuration,

$$rd = 0.059 \times 1.5 \times 10^6 \text{ bits } = 11062.5 \text{ bytes } \approx 11 \text{ packets}$$

In our simulations, the routers perform mark-tail or mark-front. The results for both strategies are compared.

6.1 Simulation Scenarios

In order to show the difference between mark-front and mark-tail strategies, we designed the following simulation scenarios based on the basic simulation model described in Figure 2. If not specified, all connections have an RTT of 59 ms, start at 0 second and stop at the 10th second.

1. One connection.
2. Two connections with the same RTT.
3. Two overlapping connections with the same RTT, but the first connection starts at 0 second and stops at the 9th second, the second connection starts at the first second and stops at the 10th second.
4. Two connections with RTT equal to 59 and 157 ms respectively.
5. Two connections with same RTT, but the buffer size at the congested router is limited to 25 packets.
6. Five connections with the same RTT.
7. Five connections with RRT of 59, 67, 137, 157 and 257 ms respectively.
8. Five connections with the same RTT, but the buffer size at the congested router is limited to 25 packets.

Scenarios 1, 4, 6 and 7 are mainly designed for testing the buffer size requirement. Scenarios 1, 3, 4, 6, 7, 8 are for link efficiency, and scenarios 2, 3, 4, 5, 6, 7 are for fairness among users.

6.2 Metrics

We use three metrics to compare the two strategies. The first metric is the *buffer size requirement* for zero loss congestion control. This is the maximum queue size that can be built up at the router in the slow start phase before the congestion signal takes effect at the congested router. If the buffer size is greater or equal to this value, no packet loss will happen. This metric is measured as the maximum queue length in the entire simulation.

The second metric, *link efficiency,* is calculated from the number of acknowledged packets (not counting the retransmissions) divided by the possible number of packets that can be transmitted during the simulated time. Because of the slow start phase and possible link idling after the window reduction, the link efficiency is always smaller than 1. Link efficiency should be measured with long simulation time to minimize the effect of the initial transient state. We tried different simulation times from 5 seconds to 100 seconds. The results for 10 seconds show the essential features of the strategy, without much difference from the results for 100 seconds. So the simulation results presented in this paper are based on 10-second simulations.

The third metric, *fairness* index, is calculated according to the formula in [17]. If m connections share the bandwidth, and x_i is the number of acknowledged packets of connection i, then the *fairness* index is calculated as:

$$fairness = \frac{(\sum_{i=1}^{m} x_i)^2}{m\sum_{i=1}^{m} x_i^2} \tag{26}$$

fairness index is often close to 1, in our graphs, we draw the $unfairness$ index:

$$unfairness = 1 - fairness. \tag{27}$$

The performance of ECN depends on the selection of the threshold value. In our results, all three metrics are drawn for different values of threshold.

6.3 Buffer Size Requirement

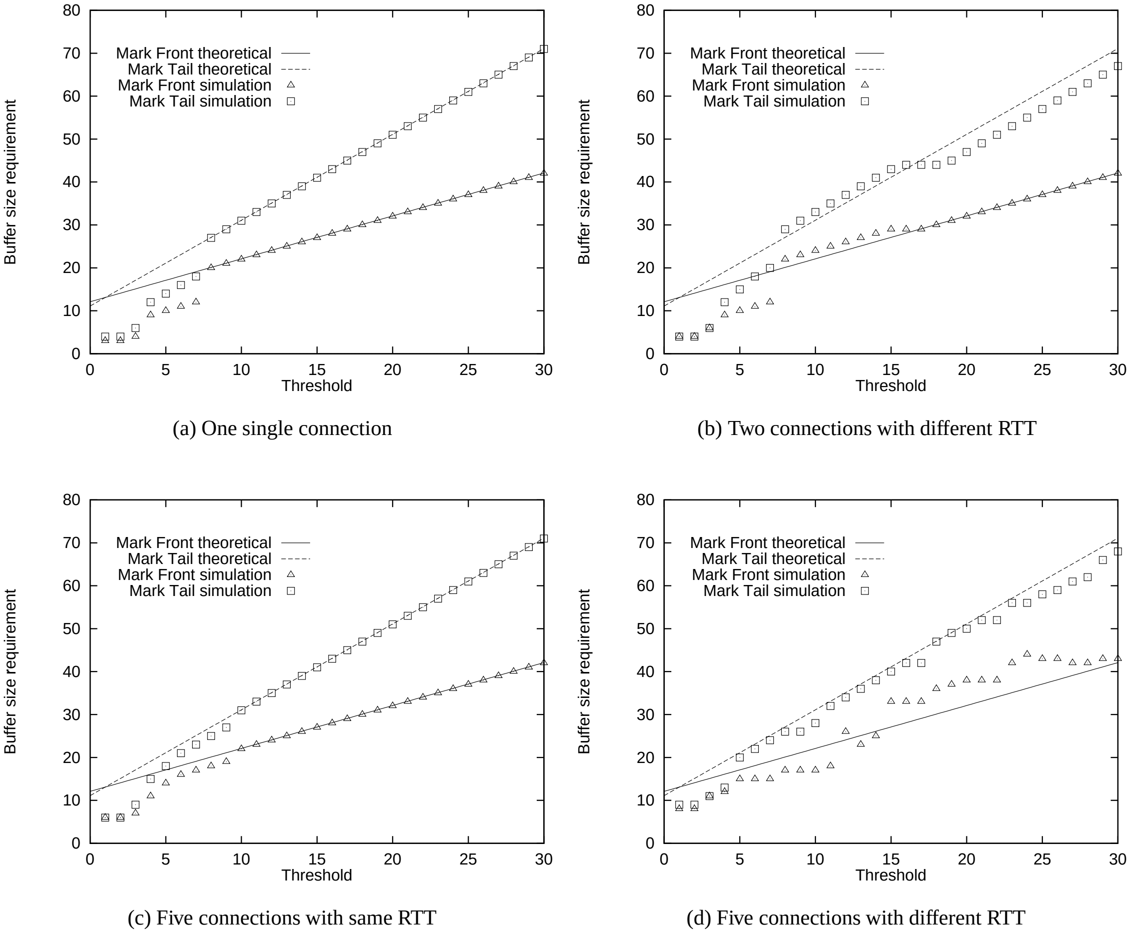
Figure 3 shows the buffer size requirement for mark-tail and mark-front. The measured maximum queue lengths are shown with "□" and "△". The corresponding theoretical estimates from Theorem 2 and 3 are shown with dashed and solid lines. In Figure 3(b) and 3(d), where the connections have different RTT, the theoretical estimate is calculated from the smallest RTT.

From the simulation, we find that for connections with the same RTT, the theoretical estimate of buffer size requirement is accurate. When threshold T is small, the buffer size requirement is an upper bound, when $T \geq rd$, the upper bound is tight. For connections with different RTT, the estimate given by the largest RTT is an upper bound, but is usually an over estimate. The estimate given by the smallest RTT is a closer approximation.

6.4 Link Efficiency

Figure 4 shows the link efficiency for various scenarios. In all cases, the efficiency increases with the threshold, until the threshold is about rd, where the link reaches almost full utilization. Small threshold results in low link utilization because it generates congestion signals even when the router is not really congested. Unnecessary window reduction actions taken by the source lead to link idling. The link efficiency results in Figure 4 verify the choice of threshold stated in Theorem 4.

In the unlimited buffer cases (a), (b), (d), (e), the difference between mark-tail and mark-front is small. However, when the buffer size is limited as in cases (c) and (f), mark-front has much better link efficiency. This is because when congestion happens, mark-front strategy provides a faster congestion feedback than mark-tail. Faster congestion



(a) One single connection

(b) Two connections with different RTT

(c) Five connections with same RTT

(d) Five connections with different RTT

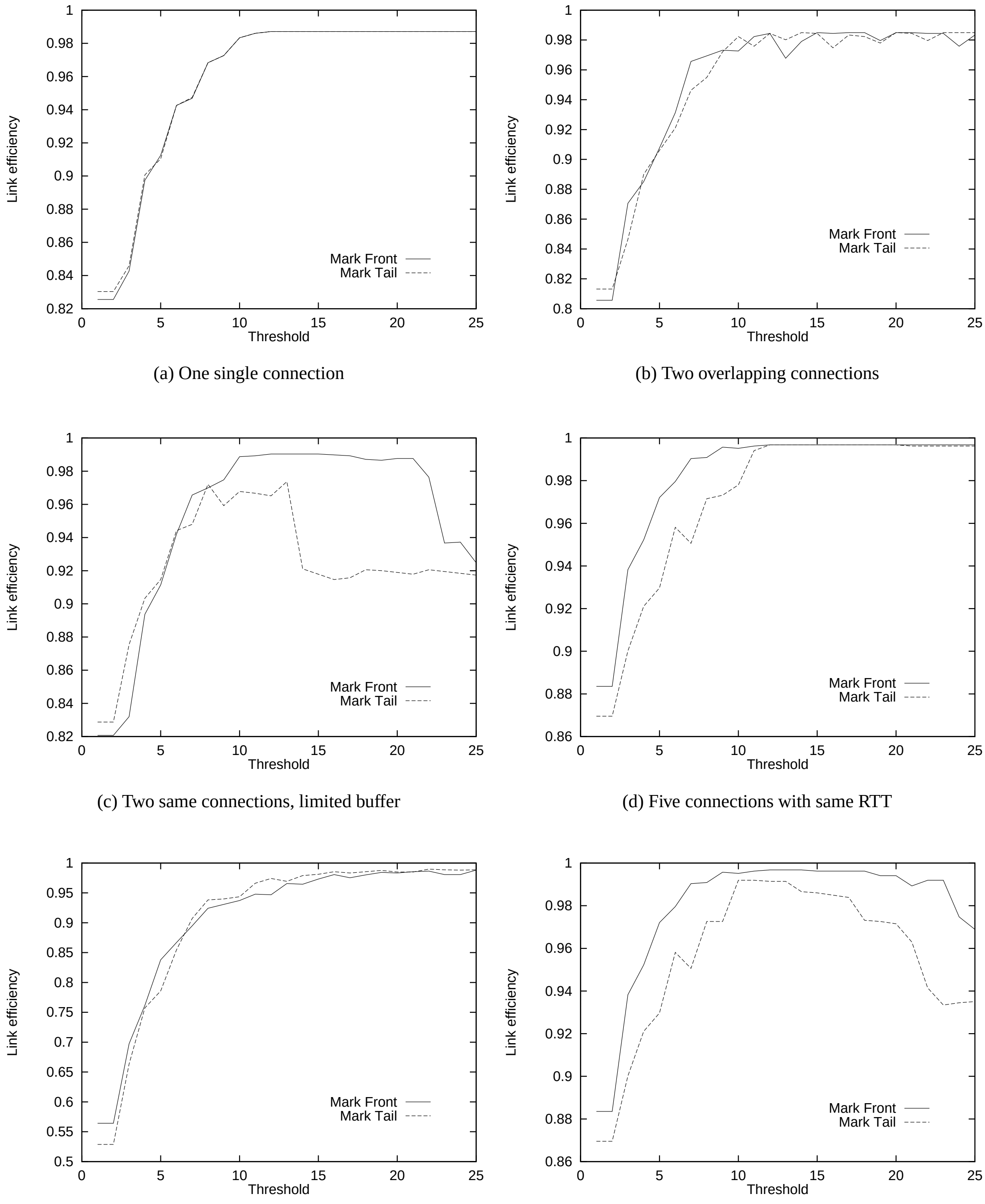
Figure 3: Buffer size requirement in various scenarios

feedback prevents the source from sending more packets that will be dropped at the congested router. Multiple drops cause source timeout and idling at the bottleneck link, and thus the low utilization. This explains the drop of link efficiency in Figure 4 (c) and (f) when the threshold exceeds about 10 packets for mark-tail and about 20 packets in mark-front.

6.5 Fairness

Scenarios 2, 3, 4, 5, 6, 7 are designed to test the scenario of the two marking strategies. Figure 5 shows lock-out phenomenon and alleviation by mark-front strategy. With the mark-tail strategy, old connections occupy the buffer and lock-out new connections. Although the two connections in scenario 3 have the same time span, the number of the acknowledged packets in the first connection is much larger than that of the second connection, Figure 5(a). In scenario 4, the connection with large RTT (157 ms) starts at the same time as the connection with small RTT (59 ms), but the connection with small RTT grows faster, takes over a large portion of the buffer room and locks out the connection with large RTT. Of all of the bandwidth, only 6.49% is allocated to the connection with large RTT. Mark-front strategy alleviates the discrimination against large RTT by marking packets already in the buffer. Simulation results show that mark-front strategy improves the portion of bandwidth allocated to connection with large RTT from 6.49% to 21.35%.

Figure 6 shows the unfairness index for the mark-tail and the mark-front strategies. In Figure 6(a), the two connections



(a) One single connection

(b) Two overlapping connections

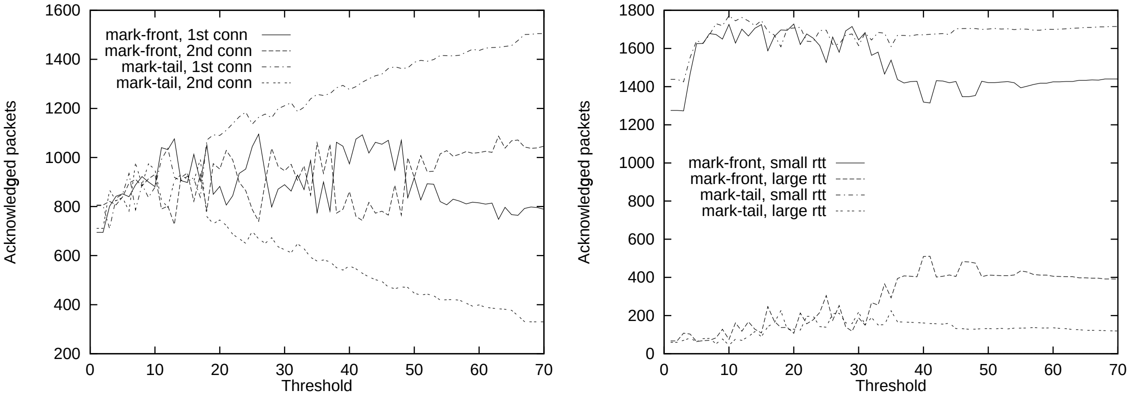
(c) Two same connections, limited buffer

(d) Five connections with same RTT

(e) Five connections with different RTT

(f) Five same connections, limited buffer

Figure 4: Link efficiency in various scenarios



(a) Two overlapping connections

(b) Two connections with different RTT

Figure 5: Lock-out phenomenon and alleviation by mark-front strategy

have the same configuration. Which connection receives more packets than the other is not deterministic, so the unfairness index seems random. But in general, mark-front has smaller unfairness index than mark-tail.

In Figure 6(b), the two connections are different: the first connection starts first and takes the buffer room. Although the two connections have the same time span, if mark-tail strategy is used, the second connection is locked out by the first and therefore receives fewer packets. Mark-front avoids this lock-out phenomenon. The results show that the unfairness index of mark-front is much smaller than that of mark-tail. In addition, as the threshold increases, the unfairness index of mark-tail increases, but the mark-front remains roughly the same, regardless of the threshold.

Figure 6(c) shows the difference on connections with different RTT. With mark-tail strategy, the connections with small RTT grow faster and therefore locked out the connections with large RTT. Since mark-front strategy does not have the lock-out problem, the discrimination against connections with large RTT is alleviated. The difference of the two strategies is obvious when the threshold is large.

Figure 6(e) shows the unfairness index when the router buffer size is limited. In this scenario, when the buffer is full, the router drops the the packet in the front of the queue. Whenever a packet is sent, the router checks whether the current queue size is larger than the threshold. If yes, the packet is marked. The figure shows that mark-front is fairer than mark-tail.

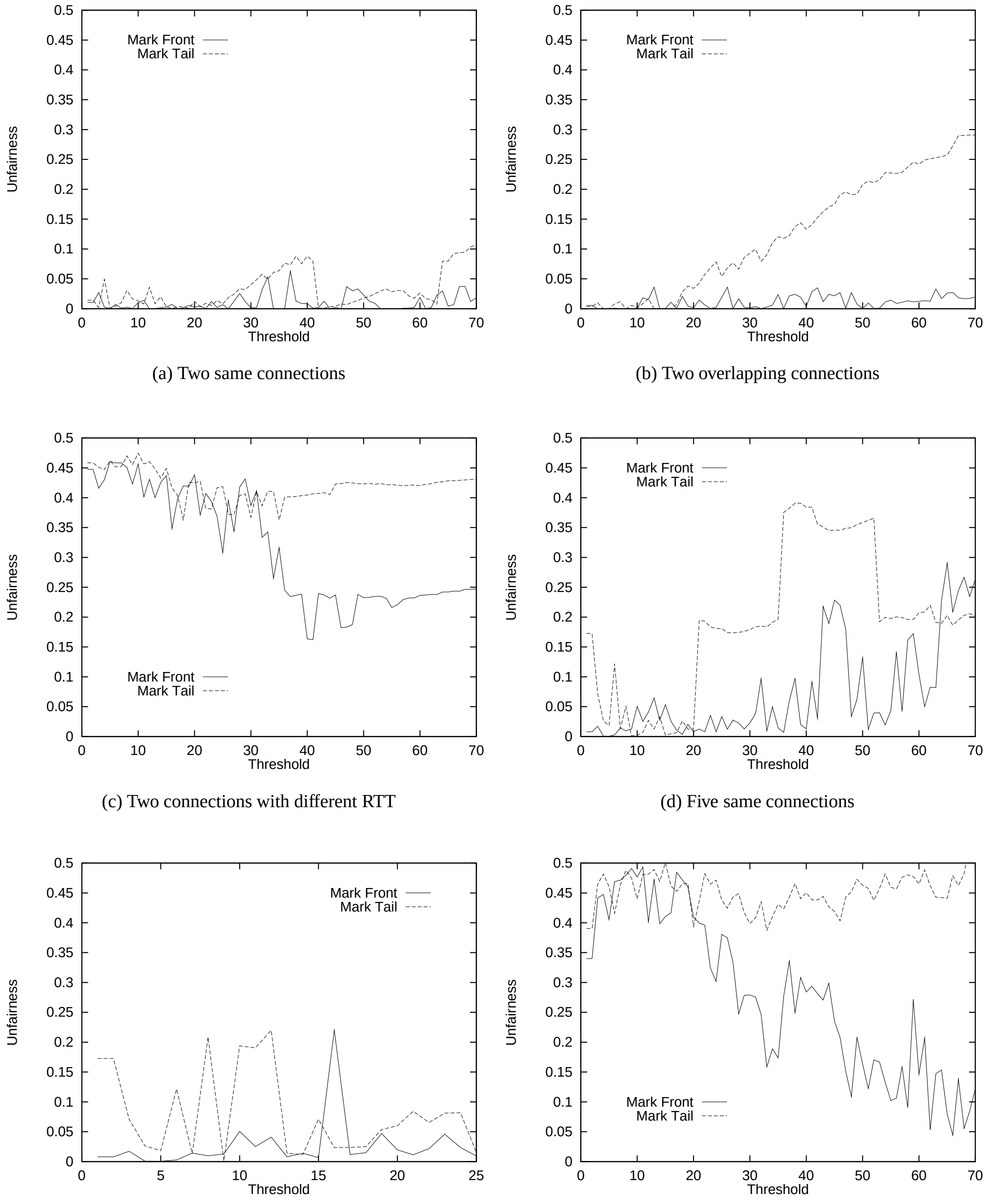
Similar results for five connections are shown in Figure 6(d) and 6(f).

7 Apply to RED

The analytical and simulation results obtained in previous sections are based on the simplified congestion detection model that a packet leaving a router is marked if the actual queue size of the router exceeds the threshold. However, RED uses a different congestion detection criterion. First, RED uses average queue size instead of the actual queue size. Second, a packet is not marked deterministically, but with a probability calculated from the average queue size.

In this section, we apply the mark-front strategy to the RED algorithm and compare the results with the mark-tail strategy. Because of the difficulty in analyzing RED mathematically, the comparison is carried out by simulations only.

RED algorithm needs four parameters: queue weight w, minimum threshold th_{min}, maximum threshold th_{max} and maximum marking probability p_{max}. Although determining the best RED parameters is out of the scope of this paper, we have tested several hundred of combinations. In almost all these combinations, mark-front has better performance than mark-tail in terms of buffer size requirement, link efficiency and fairness.



(a) Two same connections

(b) Two overlapping connections

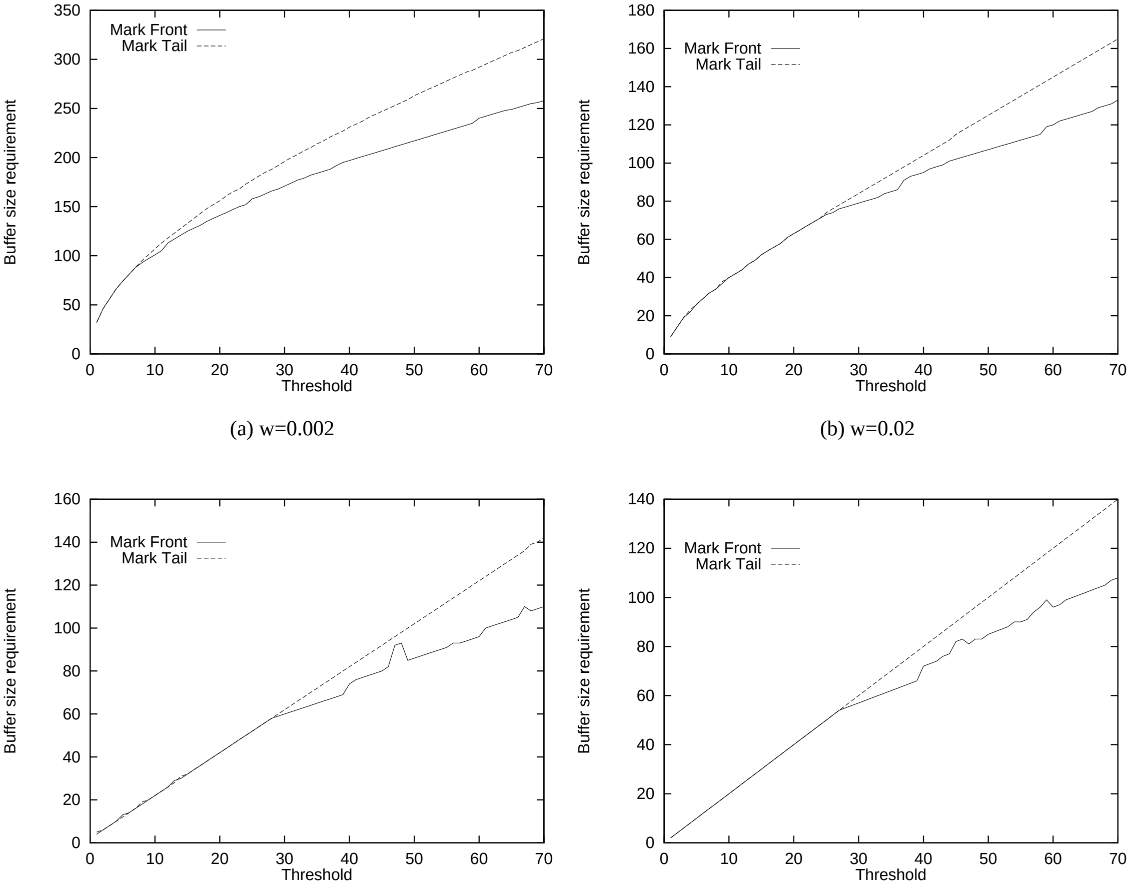
(c) Two connections with different RTT

(d) Five same connections

(e) Five same connections, limited buffer

(f) Five connections with different RTT

Figure 6: Unfairness in various scenarios



(a) w=0.002

(b) w=0.02

(c) w=0.2

(d) w=1

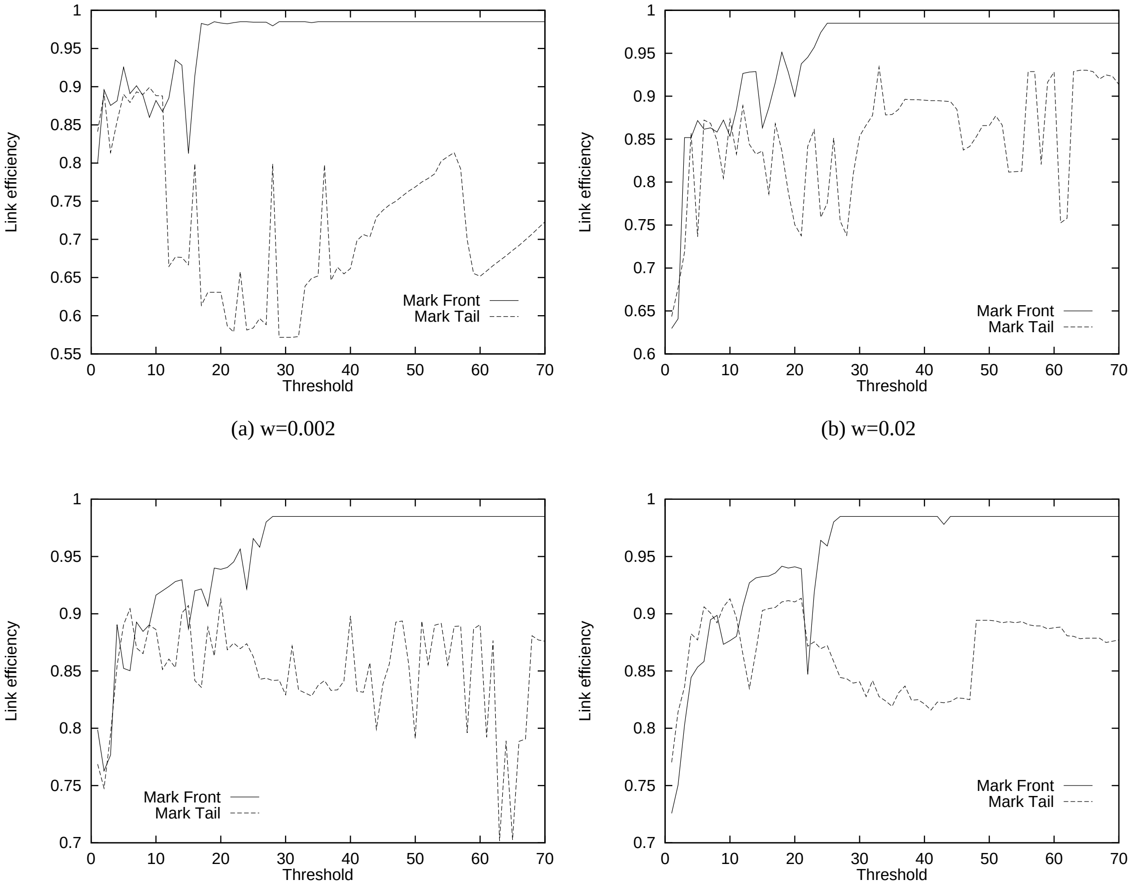
Figure 7: Buffer size requirement for different queue weight, $p_{max} = 0.1$

Instead of presenting individual parameter combinations for all scenarios, we focus on one scenario and present the results for a range of parameter values. The simulation scenario is the scenario 3 of two overlapping connections described in section 6.1. Based on the recommendations in [13], we vary the queue weight w for four values: 0.002, 0.02, 0.2 and 1, vary th_{min} from 1 to 70, fix th_{max} as $2th_{min}$, and fix p_{max} as 0.1.

Figure 7 shows the buffer size requirement for both strategies with different queue weight. In all cases, mark-front strategy requires smaller buffer size than the mark-tail. The results also show that queue weight w is a major factor affecting the buffer size requirement. Smaller queue weight requires larger buffer. When the actual queue size is used (corresponding to $w = 1$), RED requires the minimum buffer size.

Figure 8 shows the link efficiency. For almost all values of threshold, mark-front provides better link efficiency than mark-tail. Contrary to the common belief, the actual queue size (Figure 8(d)) is no worse than the average queue size (Figure 8(a)) in achieving higher link efficiency.

The queue size trace at the congested router shown in Figure 9 provides some explanation for the smaller buffer size requirement and higher efficiency of mark-front strategy. When congestion happens, mark-front delivers faster congestion feedback than mark-tail so that the sources can stop sending packets earlier. In Figure 9(a), with mark-tail signal, the queue size stops increasing at 1.98 second. With mark-front signal, the queue size stops increasing at 1.64 second. Therefore mark-front strategy needs smaller buffer.



(a) w=0.002

(b) w=0.02

(c) w=0.2

(d) w=1

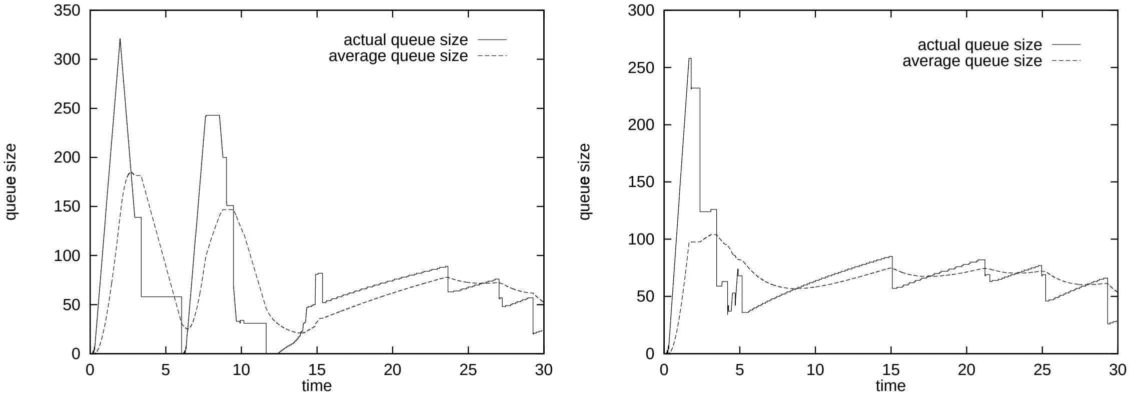
Figure 8: Link efficiency for different queue weight, $p_{max} = 0.1$

On the other hand, when congestion is gone, mark-tail is slow in reporting the change of congestion status. Packets leaving the router still carry the congestion information set at the time when they entered the queue. Even if the queue is empty, these packets still tell the sources that the router is congested. This out-dated congestion information is responsible for the link idling around 6th second and 12th second in Figure 9(a). As a comparison, in Figure 9(b), the same packets carry more up-to-date congestion information to tell the sources that the router is no longer congested, so the sources send more packets in time. Thus mark-front signal helps to avoid link idling and improve the efficiency.

Figure 10 shows the unfairness index. Both mark-front and mark-tail have big oscillations in the unfairness index when the threshold changes. These oscillations are caused by the randomness of how many packets of each connection get marked in the bursty TCP slow start phase. Changing the threshold value can significantly change the number of marked packets of each connection. In spite of the randomness, in most cases mark-front is fairer than mark-tail.

8 Conclusion

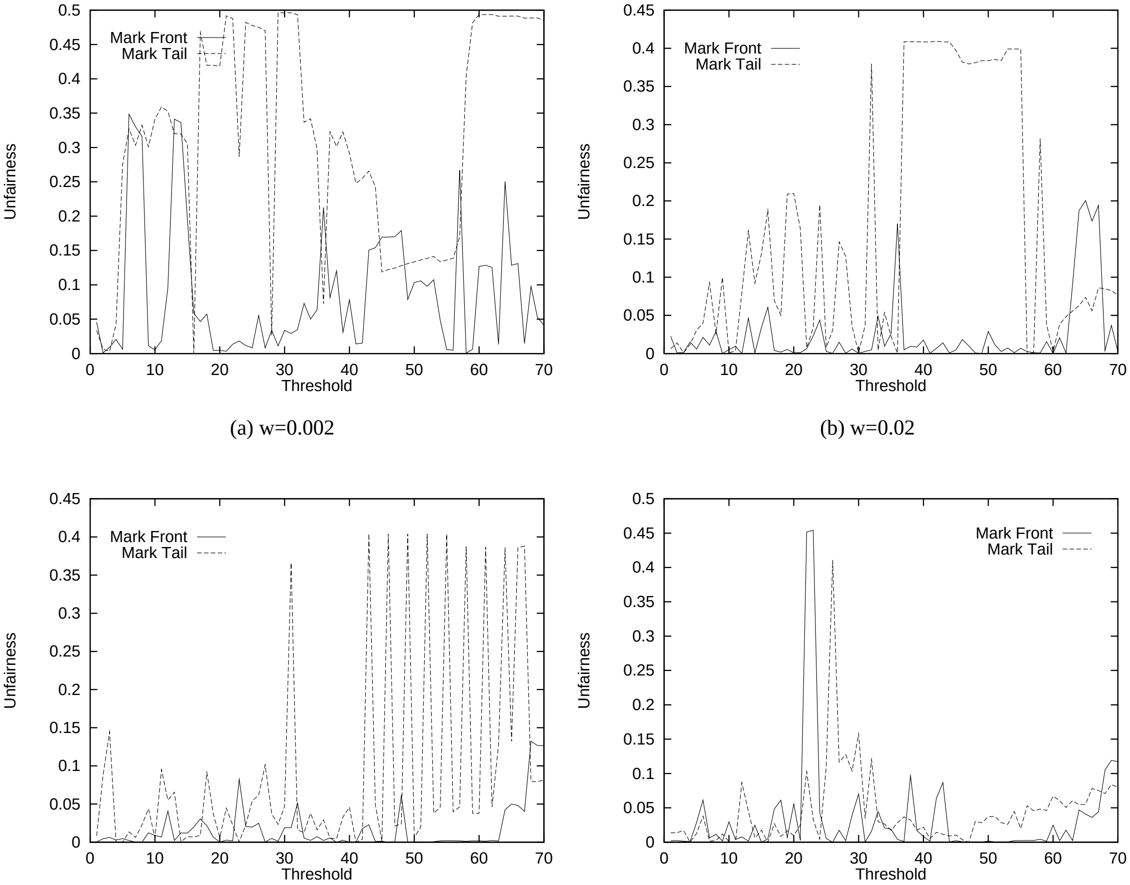
In this paper we analyze the mark-front strategy used in Explicit Congestion Notification (ECN). Instead of marking the packet from the tail of the queue, this strategy marks the packet in the front of the queue and thus delivers faster congestion signals to the source. Compared with the mark-tail policy, mark-front strategy has three advantages. First,



(a) Mark tail

(b) Mark front

Figure 9: Change of queue size at the congested router, $w = 0.002, th_{min} = 70, th_{max} = 140, p_{max} = 0.1$



(a) w=0.002

(b) w=0.02

(c) w=0.2

(d) w=1

Figure 10: Unfairness for different queue weight, $p_{max} = 0.1$

it reduces the buffer size requirement at the routers. Second, it provides more up-to-date congestion information to help the source adjust its window in time to avoid packet losses and link idling, and thus improves the link efficiency. Third, it improves the fairness among old and new users, and helps to alleviate TCP's discrimination against connections with large round trip time.

With a simplified model, we analyze the buffer size requirement for both mark-front and mark-tail strategies. Link efficiency, fairness and more complicated scenarios are tested with simulations. The results show that mark-front strategy achieves better performance than the current mark-tail policy. We also apply the mark-front strategy to the RED algorithm. Simulations show that mark-front strategy used with RED has similar advantages over mark-tail.

Based on the analysis and the simulations, we conclude that mark-front is an easy-to-implement improvement that provides a better congestion control that helps TCP to achieve smaller buffer size requirement, higher link efficiency and better fairness among users.

References

[1] V. Jacobson, Congestion avoidance and control, *Proc. ACM SIGCOMM'88*, pp. 314-329, 1988.

[2] W. Stevens, TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms, *RFC 2001*, January 1997.

[3] K. Ramakrishnan and S. Floyd, A proposal to add Explicit Congestion Notification (ECN) to IP, *RFC 2481*, January 1999.

[4] S. Floyd, TCP and explicit congestion notification, *ACM Computer Communication Review*, V. 24 N. 5, p. 10-23, October 1994.

[5] J. H. Salim, U. Ahmed, Performance Evaluation of Explicit Congestion Notification (ECN) in IP Networks, Internet Draft draft-hadi-jhsua-ecnperf-01.txt, March 2000.

[6] R. J. Gibbens and F. P. Kelly, Resource pricing and the evolution of congestion control, Automatica 35, 1999.

[7] C. Chen, H. Krishnan, S. Leung, N. Tang, Implementing Explicit Congestion Notification (ECN) in TCP for IPv6, available at http://www.cs.ucla.edu/˜tang/papers/ECN_paper.ps, Dec. 1997.

[8] UCB/LBNL/VINT Network Simulator - ns (version 2), http://www-mash.CS.Berkeley.EDU/ns/.

[9] N. Yin and M. G. Hluchyj, Implication of dropping packets from the front of a queue, *7-th ITC*, Copenhagen, Denmark, Oct 1990.

[10] T. V. Lakshman, A. Neidhardt and T. J. Ott, The drop from front strategy in TCP and in TCP over ATM, *Infocom96*, 1996.

[11] S. Floyd, RED with drop from front, email discussion on the end2end mailing list, ftp://ftp.ee.lbl.gov/email/sf.98mar11.txt, March 1998.

[12] T. V. Laksman and U. Madhow. Performance Analysis of window-based flow control using TCP/IP: the effect of high bandwidth-delay products and random loss, in *Proceedings of High Performance Networking, V. IFIP TC6/WG6.4 Fifth International Conference*, Vol C, pp 135-149, June 1994.

[13] S. Floyd and V. Jacobson, Random early detection gateways for congestion avoidance, *IEEE/ACM Transactions on Networking*, Vol. 1, No. 4, pp. 397-413, August 1993.

[14] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, An architecture for differentiated services,, *RFC 2475*, December 1998.

[15] L. Zhang, S. Shenker, and D. D. Clark, Observations and dynamics of a congestion control algorithm: the effects of two-way traffic, *Proc. ACM SIGCOMM '91*, pages 133-147, 1991.

[16] B. Braden et al, Recommendations on Queue Management and Congestion Avoidance in the Internet, *RFC 2309*, April 1998.

[17] R. Jain, *The Art of Computer System Performance Analysis*, John Wiley and Sons Inc., 1991.

Chunlei Liu received his M.Sc degree in Computational Mathematics from Wuhan University, China in 1991. He received his M.Sc degrees in Applied Mathematics and Computer and Information Science from Ohio State University in 1997. He is now a PhD candidate in Department of Computer and Information Science at Ohio State University. His research interests include congestion control, quality of service, wireless networks and network telephony. He is a student member of IEEE and IEEE Communications Society.

Raj Jain is the cofounder and CTO of Nayna Networks, Inc - an optical systems company. Prior to this he was a Professor of Computer and Information Science at The Ohio State University in Columbus, Ohio. He is a Fellow of IEEE, a Fellow of ACM and a member of Internet Society, Optical Society of America, Society of Photo-Optical Instrumentation Engineers (SPIE), and Fiber Optic Association. He is on the Editorial Boards of Computer Networks: The International Journal of Computer and Telecommunications Networking, Computer Communications (UK), Journal of High Speed Networks (USA), and Mobile Networks and Applications. Raj Jain is on the Board of Directors of MED-I-PRO Systems, LLC, Pamona, CA, and MDeLink, Inc. of Columbus, OH. He is on the Board of Technical Advisors to Amber Networks, Santa Clara, CA. Previously, he was also on the Board of Advisors to Nexabit Networks Westboro, MA, which was recently acquired by Lucent Corporation. He is also a consultant to several networking companies. For further information and publications, please see http://www.cis.ohio-state.edu/j̃ain/.

Paper 6

C. Liu and R. Jain, “Delivering Faster Congestion Feedback with the Mark-Front Strategy”

Delivering Faster Congestion Feedback with the Mark-Front Strategy [*]

Chunlei Liu Raj Jain

Department of Computer and Information Science
The Ohio State University, Columbus, OH 43210-1277, USA
email: {cliu, jain}@cis.ohio-state.edu

Abstract

Computer networks use congestion feedback from the routers and destinations to control the transmission load. Delivering timely congestion feedback is essential to the performance of networks. Reaction to the congestion can be more effective if faster feedback is provided. Current TCP/IP networks use timeout, duplicate ACKs and explicit congestion notification (ECN) to deliver the congestion feedback, each provides a faster feedback than the previous method. In this paper, we propose a mark-front strategy that delivers an even faster congestion feedback. With analytical and simulation results, we show that mark-front strategy reduces buffer size requirement, improves link efficiency and provides better fairness among users.

Keywords: Explicit Congestion Notification, mark-front, congestion control, buffer size requirement, fairness.

1 Introduction

Computer networks use congestion feedback from the routers and destinations to control the transmission load. When the feedback is "not congested", the source slowly increases the transmission window. When the feedback is "congested", the source reduces its window to alleviate the congestion [1]. Delivering timely congestion feedback is essential to the performance of networks. The faster the feedback is, the more effective the reaction to congestion can be.

TCP/IP networks uses three methods — timeout, duplicate ACKs and ECN — to deliver congestion feedback.

In 1984, Jain [2] proposed to use timeout as an indicator of congestion. When a packet is sent, the source starts a retransmission timer. If the acknowledgment is not received within a certain period of time, the source assumes congestion has happened and the packet has been lost because of the congestion. The lost packet is retransmitted and the source's congestion window is reduced. Since it has to wait for the timer to expire, timeout turns out to be the slowest feedback.

With duplicate ACKs, the receiver sends an acknowledgment after the reception of a packet. If a packet is not received but its subsequent packet arrives, the ACK for the subsequent packet is a duplicate ACK. TCP source interprets the reception of three duplicate ACKs as an indication of packet loss. Duplicate ACKs avoid the long wait for the retransmission timer to expire, and therefore, delivers a faster feedback than timeout.

Both timeout and duplicate ACKs methods send congestion feedback at the cost of packet losses, which not only increase the traffic in the network, but also add large transfer delay. Studies [3, 4, 5, 6, 7] show that the throughput of the TCP connection is limited by packet loss probability.

The congestion feedbacks from timeout and duplicate ACKs are implicit because they are inferred by the networks. In timeout method, incorrect timeout value may cause erroneous inference at the source. In duplicate ACKs method, all layers must send the packets in order. If some links have selective local link-layer retransmission, like those used in wireless links to combat transmission errors, the packets are not delivered in order. The inference of congestion from duplicate ACKs us no longer valid.

Ramakrishnan and Jain's work in [8], which has been popularly called the *DECbit scheme*, uses a single bit in the network layer header to signal the congestion. The Explicit Congestion Notification (ECN) [9, 10], motivated by the DECbit scheme, provides a mechanism for intermediate routers to send early congestion feedback to the source before actual packet losses happen. The routers monitor their queue length. If the queue length exceeds a threshold, the router marks the *Congestion Experienced* bit in the IP header. Upon the reception of a marked packet, the receiver marks the *ECN-Echo* bit in the TCP header of the acknowledgment to send the congestion feedback back to the source. In this way, ECN delivers an even faster congestion feedback explicitly set by the routers.

In most ECN implementations, when congestion happens, the congested router marks the incoming packet. When

[*] This research was sponsored in part by NSF Award #9809018 and NASA Glen Research Center.

the buffer is full or when a packet needs to be dropped as in Random Early Detection (RED), some implementations have the "drop from front" option to drop packets from the front of the queue, as suggested in Yin [12] and Lakshman [13]. However, none of these implementations mark the packet from the front of the queue.

In this paper, we propose the "mark-front" strategy. When a packet is sent from a router, the router checks whether its queue length is greater than the pre-determined threshold. If yes, the packet is marked and sent to the next router. The mark-front strategy differs from the current "mark-tail" policy in two ways. First, the router marks the packet in the front of the queue and not the incoming packet, so the congestion signal does not undergo the queueing delay as the data packets. Second, the router marks the packet at the time when it is sent, and not at the time when the packet is received. In this way, a more up-to-date congestion feedback is given to the source.

The mark-front strategy also differs from the "drop from front" option, because when packets are dropped, only implicit congestion feedback can be inferred from timeout or duplicate ACKs. When packets are marked, explicit and faster congestion feedback is sent to the source.

Our study finds that, by providing faster congestion feedback, mark-front strategy reduces the buffer size requirement at the routers; it avoids packet losses and thus improves the link efficiency when the buffer size in routers is limited. Our simulations also show that mark-front strategy improves the fairness among old and new users, and alleviates TCP's discrimination against connections with large round trip times.

This paper is organized as follows. In section 2 we describe the assumptions for our analysis. Dynamics of queue growth with TCP window control is studied in section 3. In section 4, we compare the buffer size requirement of mark-front and mark-tail strategies. In section 5, we explain why mark-front is fairer than mark-tail. In section 6, the simulation results that verify our conclusions are presented.

2 Assumptions

In [9], ECN is proposed to be used with average queue length and RED. The purpose of average queue length is to avoid sending congestion signals caused by bursty traffic, and the purpose of RED is to desynchronize sender windows [14, 15] so that the router can have a smaller queue. Because average queue length and RED are difficult to analyzed mathematically, in this paper we assume a simplified congestion detection criterion: when the *actual queue length* is smaller than the threshold, the incoming packet will not be marked; when the *actual queue length* exceeds the threshold, the incoming packet will be marked.

We also make the following assumptions. (1) Receiver windows are large enough so the bottleneck is in the network. (2) Senders always have data to send. (3) There is only one bottleneck link that causes queue buildup. (4) Receivers acknowledge every packet received and there are no delayed acknowledgments. (5) The queue length is measured in packets and all packets have the same size.

3 Queue Dynamics

In this section, we study the relationship between the window size at the source and the queue size at the congested router. The analysis is made on one connection, simulation results of multiple connections will be presented in section 6.

Under the assumption of one bottleneck, when congestion happens, packets pile up only at the bottleneck router. The following lemma is obvious.

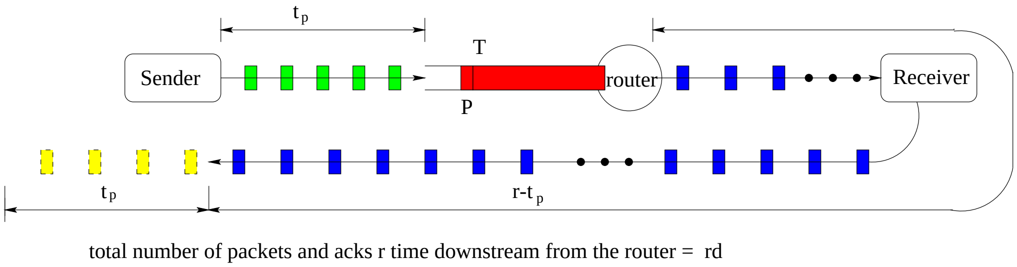
Lemma 1 *If the data rate of the bottleneck link is d packets per second, then the inter-arrival time of downstream packets and ACKs for this connection can not be shorter than $1/d$ seconds. If the bottleneck link is fully-loaded, then the inter-arrival time is $1/d$ seconds.*

Denote the source window size at time t as $w(t)$, then we have

Theorem 1 *Consider a transmission path with only one bottleneck link. Suppose the fixed round trip time is r seconds, the bottleneck link rate is d packets per second, and the propagation between the source and bottleneck router is t_p. If the bottleneck link has been busy for at least r seconds, and a packet arrives at the congested router at time t, then the queue length at the congested router is*

$$Q(t) = w(t - t_p) - rd. \qquad (1)$$

Proof Consider the packet that arrives at the congested router at time t. It was sent by the source at time $t - t_p$. At that time, the number of packets on the forward path and outstanding ACKs on the reverse path was $w(t - t_p)$. By time t, $t_p d$ ACKs are received by the source. All packets between the source and the router have entered the congested router or have been sent downstream. As shown in Figure 1, the pipe length from the congested router to the receiver, and back to the source is $r - t_p$. The number of downstream packets and outstanding ACKs are $(r - t_p)d$. The rest of the $w(t - t_p)$ unacknowledged packets are still in the congested router. So the queue length is



t_p
T
Sender
P
router
Receiver
t_p
r-t p
total number of packets and acks r time downstream from the router = rd

Figure 1: Calculation of the queue length

$$Q(t) = w(t - t_p) - t_p d - (r - t_p)d = w(t - t_p) - rd. \quad (2)$$

This finishes the proof.

Notice that in this theorem, we did not use the number of packets between the source and the congested router to estimate the queue length, because the packets downstream from the congested router and the ACKs on the reverse path are equally spaced, but the packets between the source and the congested router may not be.

4 Buffer Size Requirement

ECN feedback can be used to achieve zero-loss congestion control. If routers have enough buffer space and the threshold value is properly set, the source can control the queue length by adjusting its window size based on the ECN feedback. The buffer size requirement will be the maximum queue size that can be reached before the window reduction takes effect. In this section, we use Theorem 1 to study the buffer size requirement of mark-tail and mark-front strategies.

4.1 Mark-Tail Strategy

Suppose P is the packet that increased the queue length over the threshold T, and it was sent from the source at time s_0 and arrived at the congested router at time t_0. Its acknowledgment, which was an ECN-echo, arrived at the source at time s_1 and the window was reduced at the same time. We also assume that the last packet before the window reduction was sent at time s_1^- and arrived at the congested router at time t_1^-.

If T is reasonably large (about rd) such that the buildup of a queue of size T needs r time, the assumption in Theorem 1 is satisfied, we have

$$T = Q(t_0) = w(t_0 - t_p) - rd = w(s_0) - rd. \quad (3)$$

If T is small, rd is an overestimate of the number of downstream packets and ACKs on the reverse path. So

$$w(s_0) \leq T + rd. \quad (4)$$

Since the time elapse between s_0 and s_1 is one RTT, if packet P were not marked, the congestion window would increase to $2w(s_0)$. Because P was marked, when the ECN-Echo is received, the congestion window was

$$w(s_1^-) = 2w(s_0) - 1 \leq 2(T + rd) - 1. \quad (5)$$

When the last packet sent under this window reached the router at time t_1^-, the queue length was

$$Q(t_1^-) = w(s_1^-) - rd \leq 2T + rd - 1. \quad (6)$$

Upon the receipt of ECN-Echo, the congestion window was halved. The source can not send any more packets before half of the packets are acknowledged. So $2T + rd - 1$ is the maximum queue length.

Theorem 2 *In a TCP connection with ECN congestion control, if the fixed round trip time is r seconds, the bottleneck link rate is d packets per second, and the bottleneck router uses threshold T for congestion detection, then the maximum queue length can be reached in slow start phase is less than or equal to $2T + rd - 1$.*

When T is large, the bound $2T + rd - 1$ is tight. Since the queue length in congestion avoidance phase is smaller, this bound is actually the buffer size requirement.

4.2 Mark-Front Strategy

Suppose P is the packet that increased the queue length over the threshold T, and it was sent from the source at time s_0 and arrived at the congested router at time t_0. The router marked the packet P' that stood in the front of the queue. The acknowledgment of P', which was an ECN-echo, arrived at the source at time s_1 and the window was reduced at the same time. We also suppose the last packet before the window reduction was sent at time s_1^- and arrived at the congested router at time t_1^-.

If T is reasonably large (about rd) such that the buildup of a queue of size T needs r time, the assumption in Theorem 1 is satisfied. We have

$$T = Q(t_0) = w(t_0 - t_p) - rd = w(s_0) - rd, \qquad (7)$$

If T is small, rd is an overestimate of the number of downstream packets and ACKs on the reverse path. So

$$w(s_0) \leq T + rd. \qquad (8)$$

In slow start phase, the source increases the congestion window by one for every acknowledgment it receives. If there were no congestion, upon the reception of the acknowledgment of P, the congestion window would be doubled to $2w(s_0)$. However, when the acknowledgment of P' arrived, $T - 1$ acknowledgments corresponding to packets prior to P were still on the way. So the window size at time s_1^- was

$$w(s_1^-) = 2w(s_0) - (T - 1) - 1 \leq T + 2rd. \qquad (9)$$

When the last packet sent under this window reached the router at time t_1^-, the queue length was

$$Q(t_1^-) = w(s_1^-) - rd \leq T + 2rd - rd = T + rd. \quad (10)$$

Upon the receipt of ECN-Echo, congestion window is halved. The source can not send any more packets before half of the packets are acknowledged. So $T + rd$ is the maximum queue length.

Theorem 3 *In a TCP connection with ECN congestion control, if the fixed round trip time is r seconds, the bottleneck link rate is d packets per second, and the bottleneck router uses threshold T for congestion detection, then the maximum queue length that can be reached in slow start phase is less than or equal to $T + rd$.*

When T is large, the bound $T + rd$ is tight. Since the queue length in congestion avoidance phase is smaller, this bound is actually the buffer size requirement.

Theorem 2 and 3 estimate the buffer size requirement for zero-loss ECN congestion control. They show that the mark-front strategy reduces the buffer size requirement by rd, a bandwidth round trip time product.

5 Fairness

One of the weaknesses of mark-tail policy is its discrimination against new flows. Consider the time when a new flow joins the network and the buffer of the congested router is occupied by packets of old flows. With the mark-tail strategy, the packet that just arrived will be marked, but the packets already in the buffer will be sent without being marked. The acknowledgments of the sent packets will increase the window size of the old flows. Therefore, the old flows that already have large share of the resources will grow even larger, but the new flow with small or no share of the resources has to back off since its window size will be reduced by the marked packets. This is called a “lock-out” phenomenon because a single connection or a few flows monopolize the buffer space and prevent other connections from getting room in the queue [16]. Lock-out leads to gross unfairness among users and is clearly undesirable.

Contrary to the mark-tail policy, the mark-front strategy marks packets already in the buffer. Flows with large buffer occupancy have higher probability to be marked. Flows with smaller buffer occupancy will less likely to be marked. Therefore, old flows will back off to give part of their buffer room to the new flow. This helps to prevent the lock-out phenomenon. Therefore, mark-front strategy is fairer than mark-tail strategy.

TCP’s discrimination against connections with large RTTs is also well known. The cause of this discrimination is similar to the discrimination against new connections. Connections with small RTTs receives their acknowledgment faster and therefore grow faster. Starting at the same time as connections with large RTTs, connections with small RTTs will take larger room in the buffer. With mark-tail policy, packets already in the queue will not be marked but only newly arrived packets will be marked. Therefore, connections with small RTTs will grow even larger, but connections with large RTTs have to back off. Mark-front alleviates this discrimination by treating all packets in the buffer equally. Packets already in the buffer may also be marked. Therefore, connections with large RTTs can have larger bandwidth.

6 Simulation Results

In order to compare the mark-front and mark-tail strategies, we performed a set of simulations with the ns simulator [11].

6.1 Simulation Models

Our simulations are based on the basic simulation model shown in Figure 2. A number of sources $s_1, s_2, \ldots, s_m$ are connected to the router r_1 by 10 Mbps links. Router r_1 is connected to r_2 by a 1.5 Mbps link. Destinations $d_1, d_2, \ldots, d_m$ are connected to r_2 by 10 Mbps links. The

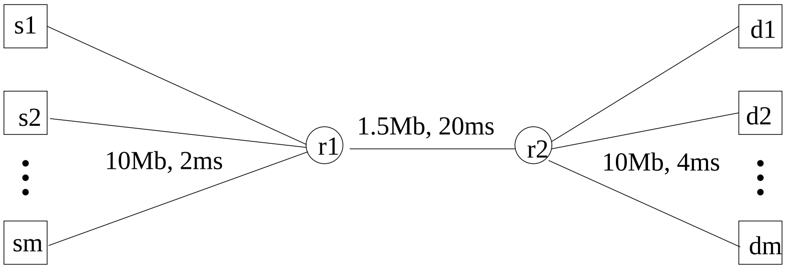


Figure 2: Simulation model.

link speeds are chosen so that congestion will only happen at the router r_1, where mark-tail and mark-front strategies are tested.

With the basic configuration, the fixed round trip time, including the propagation time and the processing time at the routers, is 59 ms. Changing the propagation delay between router r_1 and r_2 from 20 ms to 40 ms gives an RTT of 99 ms. Changing the propagation delays between the sources and router r_1 gives us configurations of different RTTs. An FTP application runs on each source. The data packet size is 1000 bytes and the acknowledgment packet size is 40 bytes. TCP Reno and ECN are used for congestion control.

The following simulation scenarios are designed on the basic simulation model. In each of the scenarios, if not otherwise specified, all connections have an RTT of 59 ms, start at 0 second and stop at the 10th second.

1. One single connection.
2. Two connections with the same RTT, starting and ending at the same time.
3. Two connections with the same RTT, but the first connection starts at 0 second and stops at the 9th second, the second connection starts at the first second and stops at the 10th second.
4. Two connections with RTT equal to 59 and 157 ms respectively.
5. Two connections with same RTT, but the buffer size at the congested router is limited to 25 packets.
6. Five connections with the same RTT.
7. Five connections with RRT of 59, 67, 137, 157 and 257 ms respectively.
8. Five connections with the same RTT, but the buffer size at the congested router is limited to 25 packets.

Scenarios 1, 4, 6 and 7 are mainly designed for testing the buffer size requirement. Scenarios 1, 3, 4, 6, 7, 8 are for link efficiency, and scenarios 2, 3, 4, 5, 6, 7 are for fairness among users.

6.2 Metrics

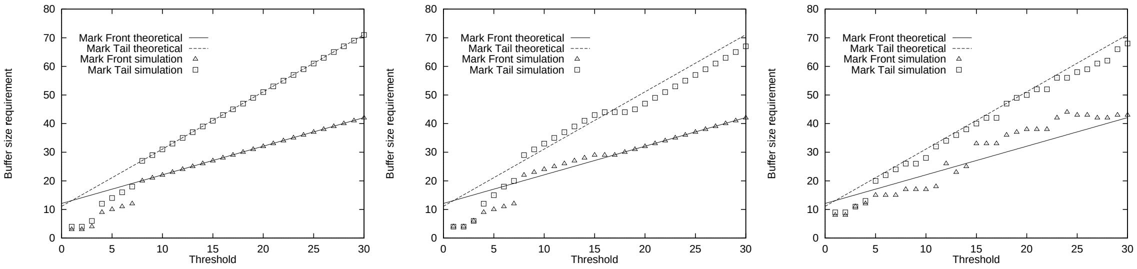
We use three metrics to compare the the results. The first metric is the *buffer size requirement* for zero loss congestion control, which is the maximum queue size that can be built up at the router in the slow start phase before the congestion feedback takes effect. If the buffer size is greater or equal to this value, the network will not suffer packet losses. The analytical results for one connection are given in Theorem 2 and 3. Simulations will be used in multiple-connection and different RTT cases.

The second metric, *link efficiency*, is calculated from the number of acknowledged packets and the possible number of packets that can be transmitted during the simulation time. There are two reasons that cause the link efficiency to be lower than full utilization. The first reason is the slow start process. In the slow start phase, the congestion window grows from one and remains smaller than the network capacity until the last round. So the link is not fully used in slow start phase. The second reason is low threshold. If the congestion detection threshold T is too small, ECN feedback can cause unnecessary window reductions. Small congestion window leads to link under-utilization. Our experiments are long enough so that the effect of the slow start phase can be minimized.

The third metric, *fairness* index, is calculated according to the method described in [17]. If m connections share the bandwidth and x_i is the throughput of connection i, the *fairness* index is calculated as:

$$fairness = \frac{(\sum_{i=1}^{m} x_i)^2}{m \sum_{i=1}^{m} x_i^2} \qquad (11)$$

When all connections have the same throughput, the fairness index is 1. The farther the throughput distribution is away from the equal distribution, the smaller the fairness value is. Since the fairness index in our results is often



(a) One single connection

(b) Two connections with different RTTs

(c) Five connections with different RTTs

Figure 3: Buffer size requirement in various scenarios

close to 1, in our graphs, we draw the $unfairness$ index:

$$unfairness = 1 - fairness, \tag{12}$$

to better contrast the difference.

The operations of ECN depend on the threshold value T. In our results, all three metrics are drawn for different values of threshold.

6.3 Results

Figure 3 shows the buffer size requirement for mark-tail and mark-front strategies. The measured maximum queue lengths are shown with "□" and "△". The corresponding analytical estimates from Theorem 2 and 3 are shown with dashed and solid lines. Figure 3(a) shows the buffer size requirement for one single connection with an RTT of 59 ms. Figure 3(b) shows the requirement for two connections with different RTTs. Figure 3(c) shows the requirement for five connections with different RTTs. When the connections have different RTTs, the analytical estimate is calculated from the smallest RTT.

From these results, we find that for connections with equal RTTs, the analytical estimate of buffer size requirement is accurate. When threshold T is small, the buffer size requirement is an upper bound, when $T \geq rd$, the upper bound is tight. For connections with different RTTs, the estimate given by the largest RTT is an upper bound, but is usually an overestimate. The estimate given by the smallest RTT is a closer approximation.

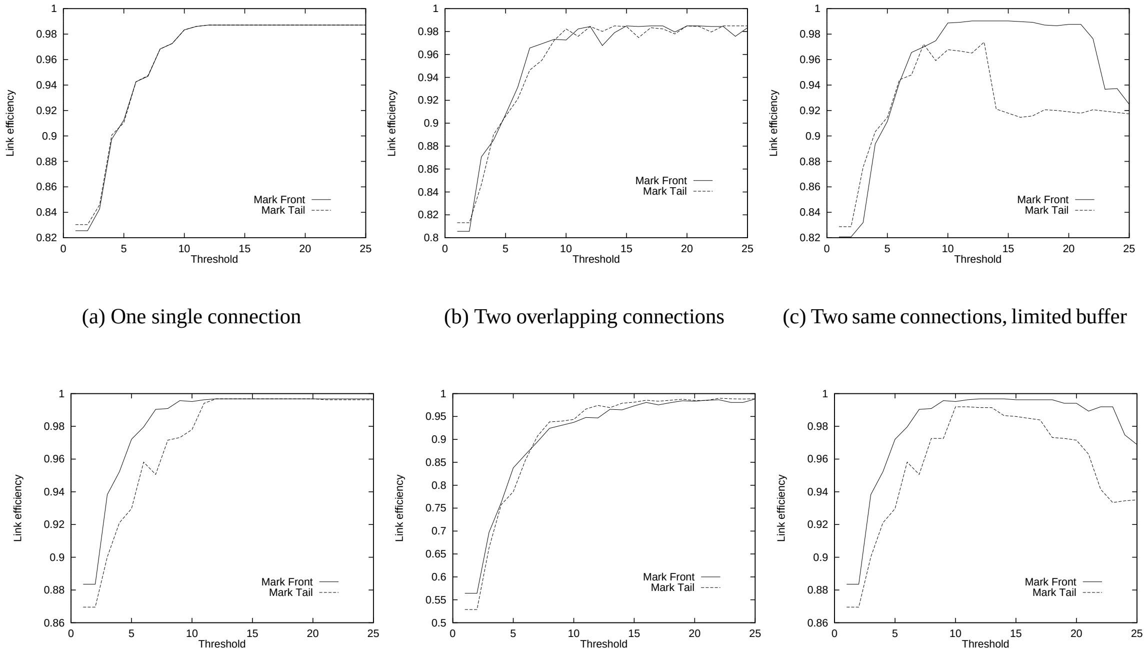
Figure 4 shows the link efficiency. Results for mark-front strategy are drawn with solid line, and results for mark-tail strategy are drawn with dashed line. In most cases, when the router buffer size is large enough, mark-front and mark-tail have comparable link efficiency, but when the threshold is small, mark-front have slightly lower efficiency because congestion feedback is sent to the source faster. For the same value of threshold, faster feedback translates to more window reductions and longer link idling.

When the router buffer size is small, as in Figure 4(c) and Figure 4(f), mark-front has better link efficiency. This is because mark-front sends congestion feedback to source faster, so the source can reduce its window size sooner to avoid packet losses. Without spending time on the retransmissions, mark-front strategy can improve the link efficiency.

Figure 5 shows the unfairness. Again, results for mark-front strategy are drawn with solid line, and results for mark-tail strategy are drawn with dashed line. In Figure 5(a), the two connections have the same configuration. Which connection receives more packets than the other is not deterministic, so the unfairness index seems random. However, in general, mark-front is fairer than mark-tail.

In Figure 5(b), the two connections are different: the first connection starts first, occupies the buffer room and locks out the second connection. Although they have the same time span, the second connection receives fewer packets than the first. Mark-front avoids this lock-out phenomenon and improves the fairness. In addition, as the threshold increases, the unfairness index of mark-tail increases, but the mark-front remains roughly the same, regardless of the threshold. Results for five same connections are shown in Figure 5(d).

Figure 5(c) shows the difference on connections with different RTTs. With mark-tail strategy, the connections with small RTTs grow faster and therefore locked out the connections with large RTTs. Mark-front strategy avoids the lock-out problem and alleviate the discrimination against connections with large RTT. The difference of the two strategies is obvious when the threshold is large. Results for five connections with different RTTs are shown in Figure 5(f).



(a) One single connection

(b) Two overlapping connections

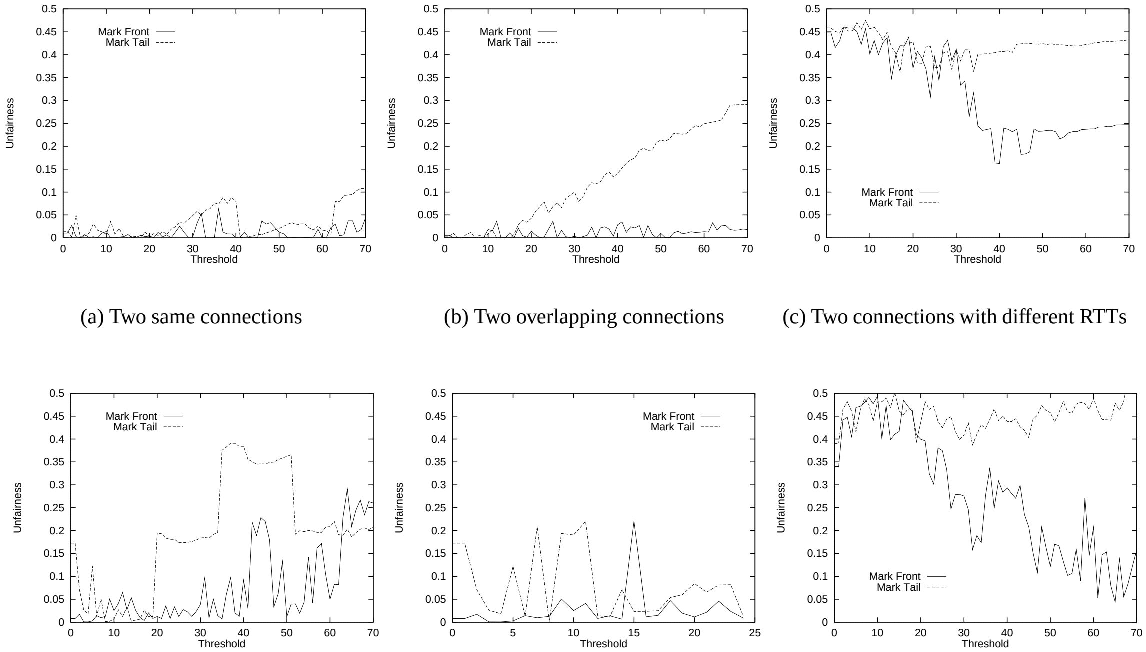
(c) Two same connections, limited buffer

(d) Five connections with same RTT

(e) Five connections with different RTTs

(f) Five same connections, limited buffer

Figure 4: Link efficiency in various scenarios



(a) Two same connections

(b) Two overlapping connections

(c) Two connections with different RTTs

(d) Five same connections

(e) Five same connections, limited buffer

(f) Five connections with different RTTs

Figure 5: Unfairness in various scenarios

Figure 5(e) shows the unfairness when the router buffer size is limited. In this scenario, the mark-tail strategy marks the incoming packet when the queue length exceeds the threshold, and drops the incoming packet when the buffer is full. The mark-front strategy, on the other hand, marks and drops the packets from the front of the queue when necessary. The results show mark-front strategy is fairer than mark-tail.

7 Conclusion

In this paper we study the mark-front strategy used in ECN. Instead of marking the packet from the tail of the queue, this strategy marks the packet in the front of the queue and thus delivers faster congestion feedback to the source. Our study reveals mark-front's three advantages over mark-tail policy. First, it reduces the buffer size requirement at the routers. Second, when the buffer size is limited, it reduces packet losses and improves the link efficiency. Third, it improves the fairness among old and new users, and helps to alleviate TCP's discrimination against connections with large round trip times.

With a simplified model, we analyze the buffer size requirement for both mark-front and mark-tail strategies. Link efficiency, fairness and more complicated scenarios are tested with simulations. The results show that mark-front strategy has better performance than the current mark-tail policy.

References

[1] W. Stevens, TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms, *RFC 2001*, January 1997.

[2] R. Jain, A Timeout-Based Congestion Control Scheme for Window Flow-Controlled Networks, *IEEE Journal on Selected Areas in Communications*, Vol. SAC-4, No. 7, pp. 1162-1167, October 1986.

[3] S. Floyd, Connections with multiple congestion gateways in packet-switched networks: part 1: one-way traffic, *Computer Communication Review*, 21(5), October 1991.

[4] T. V. Lakshman, U. Madhow, Performance analysis of window-based flow control using TCP/IP: effect of high bandwidth-delay products and random loss, *IFIP Transactions C: Communication Systems*, C-26, pp.135-149, 1994.

[5] M. Mathis, J. Semke, J. Mahdavi, T. Ott, The macroscopic behavior of the TCP congestion avoidance algorithm, *Computer Communication Review*, volume 27, number3, July 1997.

[6] T. Ott, J. Kemperman, and M. Mathis, The stationary behavior of ideal TCP congestion avoidance, ftp://ftp.bellcore.com/pub/tjo/TCPwindow.ps, August 1996.

[7] J. Padhye, V. Firoiu, D. Towsley, J. Kurose, Modeling TCP throughput: a simple model and its empirical validation, *Computer Communication Review*, 28(4), pp. 303-314, 1998.

[8] K. Ramakrishnan and R. Jain, A binary feedback scheme for congestion avoidance in computer networks , *ACM Transactions on Computer Systems*, Vol. 8, No. 2, pp. 158-181, May 1990.

[9] K. Ramakrishnan and S. Floyd, A proposal to add Explicit Congestion Notification (ECN) to IP, *RFC 2481*, January 1999.

[10] S. Floyd, TCP and explicit congestion notification, *ACM Computer Communication Review*, V. 24 N. 5, pp. 10-23, October 1994.

[11] UCB/LBNL/VINT Network Simulator - ns (version 2), http://www-mash.CS.Berkeley.EDU/ns/.

[12] N. Yin and M. G. Hluchyj, Implication of dropping packets from the front of a queue, 7-th ITC, Copenhagen, Denmark, Oct 1990.

[13] T. V. Lakshman, A. Neidhardt and T. J. Ott, The drop from front strategy in TCP and in TCP over ATM, Infocom96, 1996.

[14] S. Floyd and V. Jacobson, Random early detection gateways for congestion avoidance, *IEEE/ACM Transactions on Networking*, Vol. 1, No. 4, pp. 397-413, August 1993.

[15] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang and W. Weiss, An architecture for differentiated services,, *RFC 2475*, December 1998.

[16] B. Braden et al, Recommendations on Queue Management and Congestion Avoidance in the Internet, *RFC 2309*, April 1998.

[17] R. Jain, *The Art of Computer System Performance Analysis*, John Wiley and Sons Inc., 1991.

All our papers and ATM Forum contributions are available through http://www.cis.ohio-state.edu/˜jain/

Mid Term Presentation
at
NASA Glenn Research Center,
Cleveland, Ohio.

June 2, 2000

QoS for Real Time Applications over Next Generation Data Networks

Project Status Report: Part 1
June 2, 2000

http://www.engr.udayton.edu/faculty/matiquzz/Pres/QoS.pdf

University of Dayton
Mohammed Atiquzzaman
Jyotsna Chitri
Faruque Ahamed
Hongjun Su
Haowei Bai

Ohio State University
Raj Jain
Mukul Goyal
Bharani Chadavala
Arindam Paul
Chun Lei Liu
Wei Sun
Arian Durresi

NASA Glenn
Will Ivancic

Outline of Talk

- Present state of the Interent.
- QoS approaches to future data networks.
- Research Issues.
- Progress to date:
  - Task 1: DS over ATM
  - Task 2: IS over DS
  - Task 3: ATN
  - Task 4: Satellite networks
  - Task 5: MPLS
- Conclusions.

Current Internet

- TCP/IP glues together all the computers in the Internet.
- TCP/IP was designed for terrestrial networks.
- TCP/IP does not
  - offer QoS to real time applications, or
  - perform well in long delay bandwidth networks.

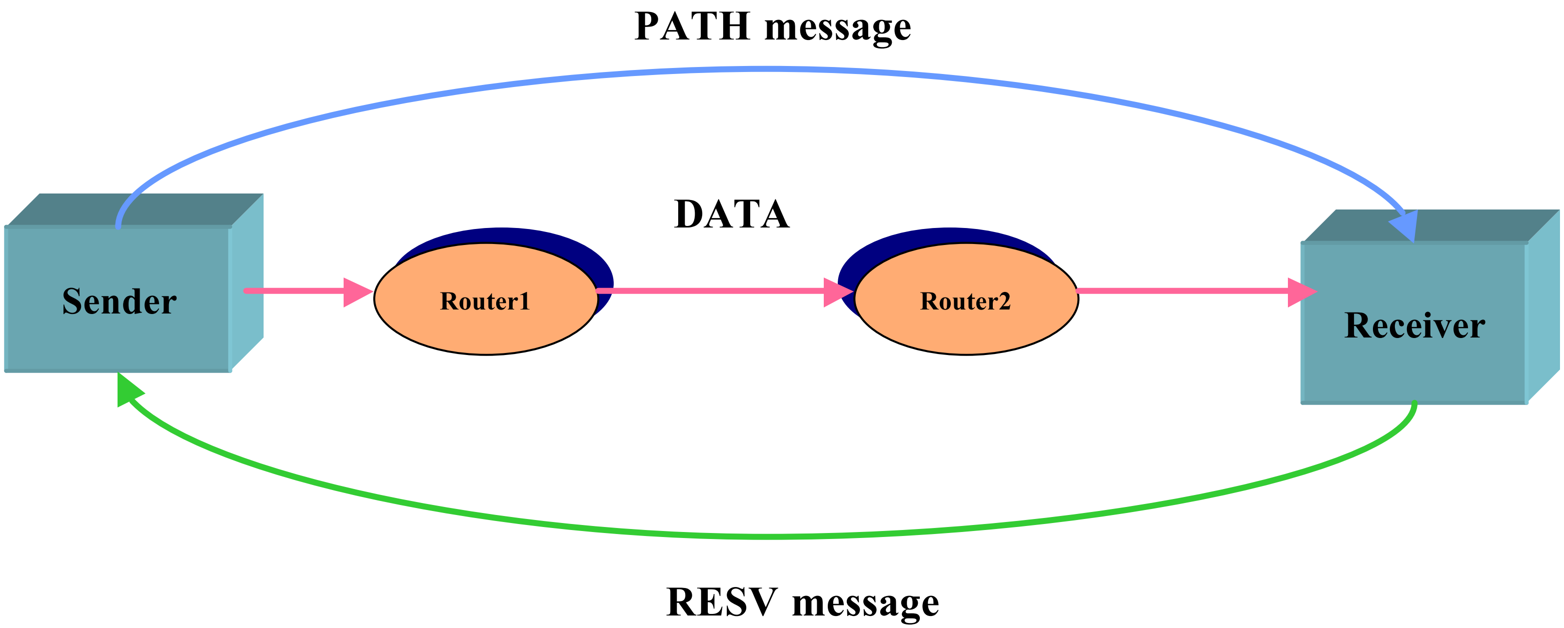
Efforts to provide QoS in Internet

- Integrated Services (IS)
- Differentiated Services (DiffServ)
- Explicit Congestion Notification (ECN)
- Multiprotocol Label Switching (MPLS)
- Asynchronous Transfer Mode (ATM)

Integrated Services

- RSVP to reserve resources during connection setup.
- End-to-end QoS guarantees.
- A router has to keep information about all connections passing through the router.
- Gives rise to scalability problem in the core routers.

RSVP Signaling



RSVP Signaling

- Reserves a portion of link bandwidth in each router
- The sender sends a PATH message with resource requirements for a flow.
- Receiver responds with a RESV message
- Each router processes the RESV to reserve the required resources requested by the sender.
- Routers can modify the QoS parameters of the RESV message if enough resources are not available to meet the requirements.
- Each router in the entire path confirms the end-to-end reservation for the flow.

IS Service Classes

- **Guaranteed Load Service**
  - **Low end-to-end delay, Jitter, Loss.**
  - **Highest priority service.**
- **Controlled Load Service**
  - **Network should forward the packets with queuing delay not greater than that caused by the traffic's own burstiness (RFC 2474).**
  - **Performance similar to that of an unloaded network.**
  - **Traffic specifications from the Tspec.**
- **Best Effort**

Differentiated Services

- Similar traffic are grouped into classes.
- Resources reserved for classes.
- QoS provided to classes.
  - QoS to individual connections is an open research issue.
- QoS maintained by:
  - Classification
  - Traffic policing
    - Metering, dropping, tagging
  - Traffic shaping
- Per Hop Behavior (PHB)
  - Specifies QoS received by packets i.e. how packets are treated by the routers.

Asynchronous Transfer Mode

- Strong QoS guarantees; suitable for real time applications.
- High cost prohibits use at the edge network or to the desktop.
- Currently used at the core of the Internet.

Next Generation Internet

- Routers at the edge network will not need to carry too many connections
  - IS can be used at the edge network.
- Core network needs to carry lot of connections.
  - Combination of DS, ATM and MPLS at the core.
- Satellite/Wireless links
  - Remote connectivity and mobility.

Research Issues

- Service mapping between networks.
- Loss and delay guarantees.
- Interoperability among edge and core technologies.
- Interoperability with Aeronautical Telecommunications Network (ATN).
- Operation in satellite environment having high delay and loss.

Task 1

Prioritized Early Packet Discard

Asynchronous Transfer Mode (ATM)

- Service Classes
  - Constant Bits Rate (CBR)
  - Available Bit Rate (ABR)
  - Unspecified Bit Rate (UBR)
- CLP in cell header
  - Determines loss priority of packets

Differentiated Services

- Service Classes
  - Premium Service: emulates leased line
  - Assured Service
  - Best Effort Service
- Various levels of drop precedence.
  - Need to be mapped to ATM when running DS over ATM.
  - Could be possible mapped to the CLP bit of ATM cell header.

DS over ATM

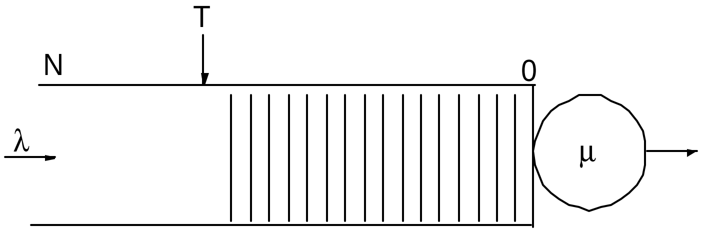
- Possible Service Mappings
  - Premium Service → ATM CBR service.
  - Assured Service → ATM UBR service with CLP=0
  - Best Effort → ATM UBR service with CLP=1
- DS packets are broken down into cells at the DS-ATM gateway
  - Drop precedence mapped to CLP bit
- Buffer Management at ATM switches
  - Partial Packet Discard (PPD)
  - Early Packet Discard (EPD)

Prioritized EPD

- DS service classes can use the CLP bit of ATM cell header to provide service differentiation.
- EPD does not consider the priority of cells.
- Prioritized EPD can be used to provide service discrimination.
- Two thresholds are used to drop cells depending on the CLP bit.

Buffer Management Schemes

EPD



- QL < T

  Accept all packets.

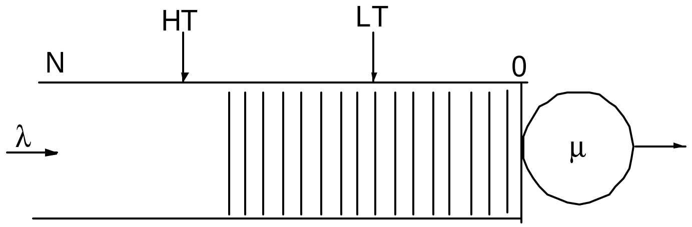
- T ≤ QL < N

  Discard all new incoming packets.

- QL ≥ N

  - Discard all.

PEPD



- QL <LT

  Accept all packets.

- LT ≤ QL < HT

  Discard all new low priority packets.

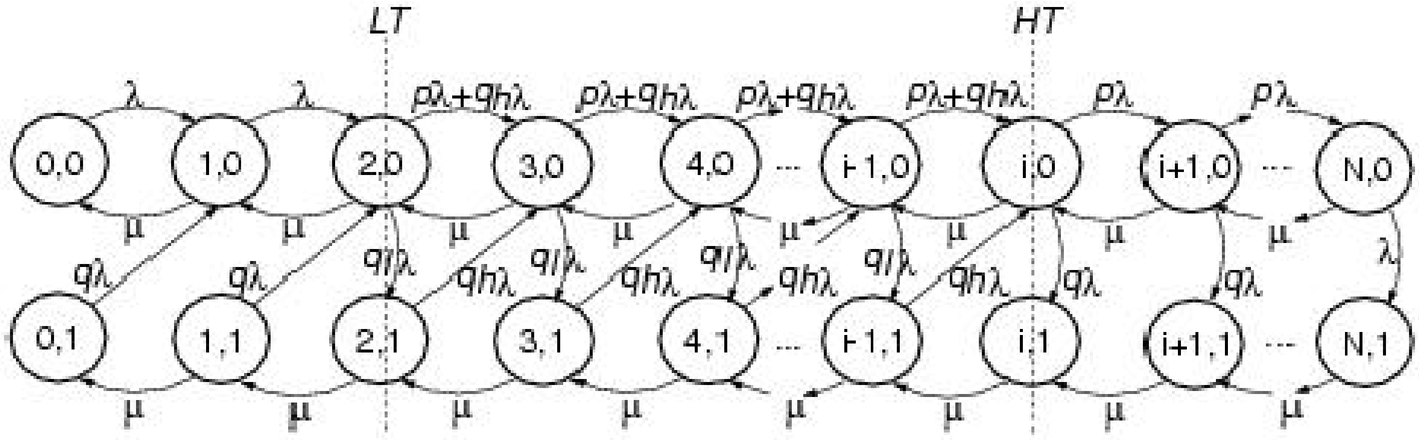
- HT ≤ QL < N

  Discard all new packets

QL ≥ N

- Discard all packets

Steady State Diagram



Steady State Equations

$$l\ P_{0,0} = m\,P_{1,0}$$

$$q\,l\ \ P_{0,1} = m\,P_{1,1}$$

$$(l\ + m)P_{i,0} = l\ \ P_{i-1,0} + m\,P_{i+1,0} + q\,l\ \ P_{i-1,1} \qquad 1 \le i \le LT$$

$$(l\ + m)P_{i,0} = (l\ \ p + qh\,l\ )\,P_{i-1,0} + m\,P_{i+1,0} + qh\,l\ P_{i-1,1} \qquad LT < i \le HT$$

$$(l\ + m)P_{i,0} = p\ l\ \ P_{i-1,0} + m\,P_{i+1,0} \qquad HT < i < N$$

$$(l\ + m)P_{N,0} = p\ l\ \ P_{N-1,0}$$

$$m\ \ P_{N,1} = l\ \ P_{N,0}$$

$$m\ \ P_{i,1} = q\,l\ \ P_{i,0} + m\,P_{i+1,1} \qquad HT \le i < N$$

$$(m + qh\,l\ )P_{i,1} = q\,l\ \ (1-h)\ P_{i,0} + m\,P_{i+1,1} \qquad LT \le i < HT$$

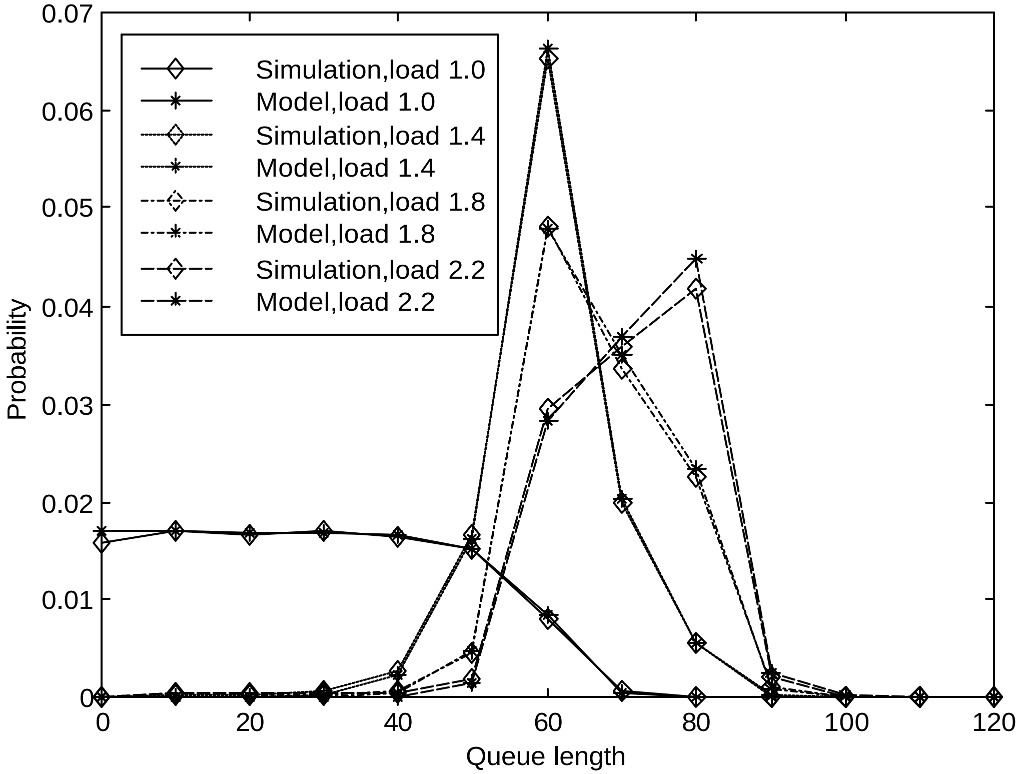
$$(m + q\,l\ )P_{i,1} = m\,P_{i+1,1} \qquad 0 < i < LT$$

$$\sum_{i=0}^{N}(P_{i,0} + P_{i,1}) = 1$$

Goodput

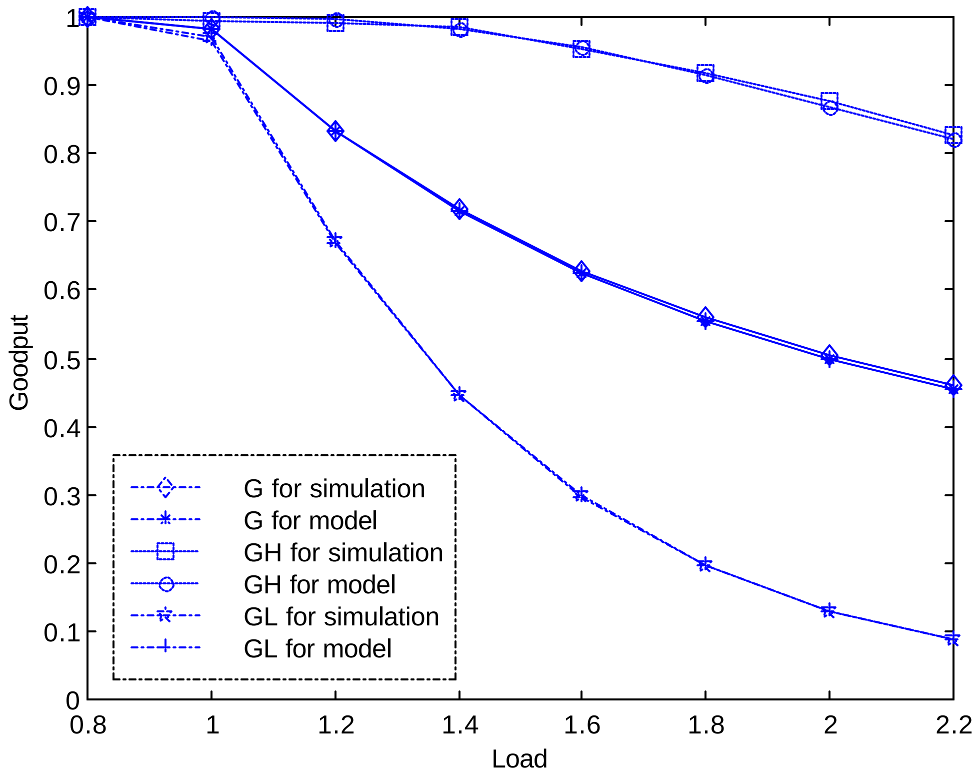
$$G_h = \frac{\sum_{n=1}^{\infty} nP(W = n, V = 1, U = 1)}{\sum_{n=1}^{\infty} nP(W = n, U = 1)}$$

Queue Occupancy



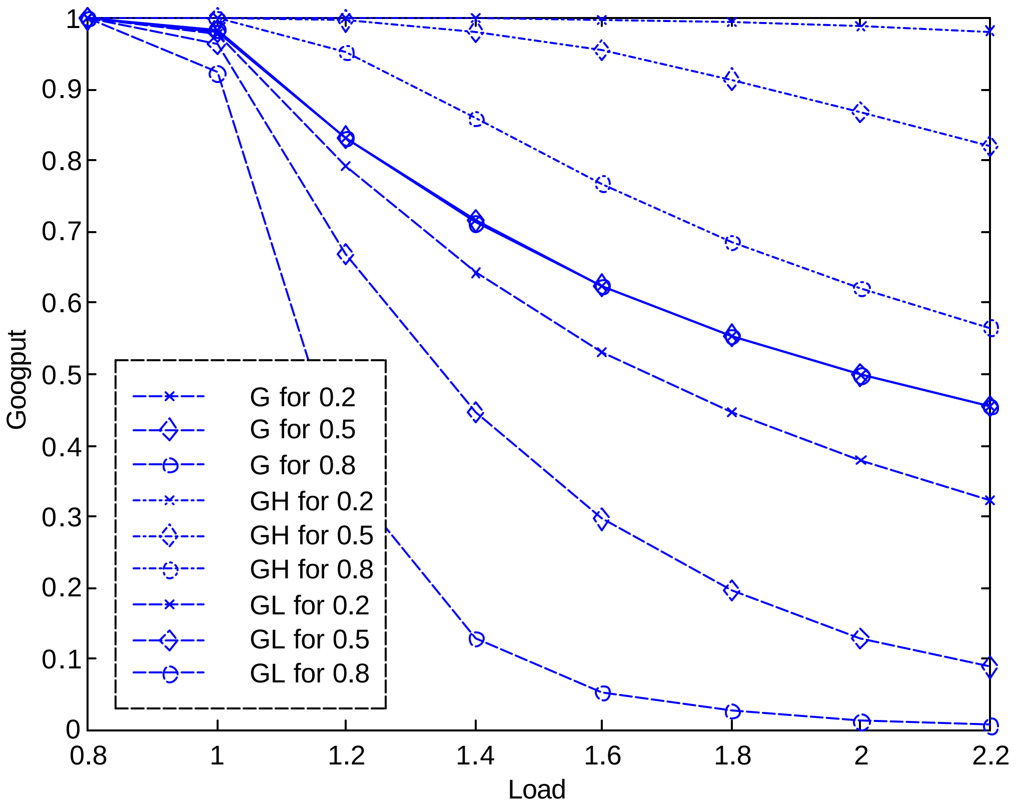
***Simulation:* *N*=120, *LT*=60, *HT*=80, *h*=0.5, *q*=1/6.**

Goodput versus load for *h*=0.5



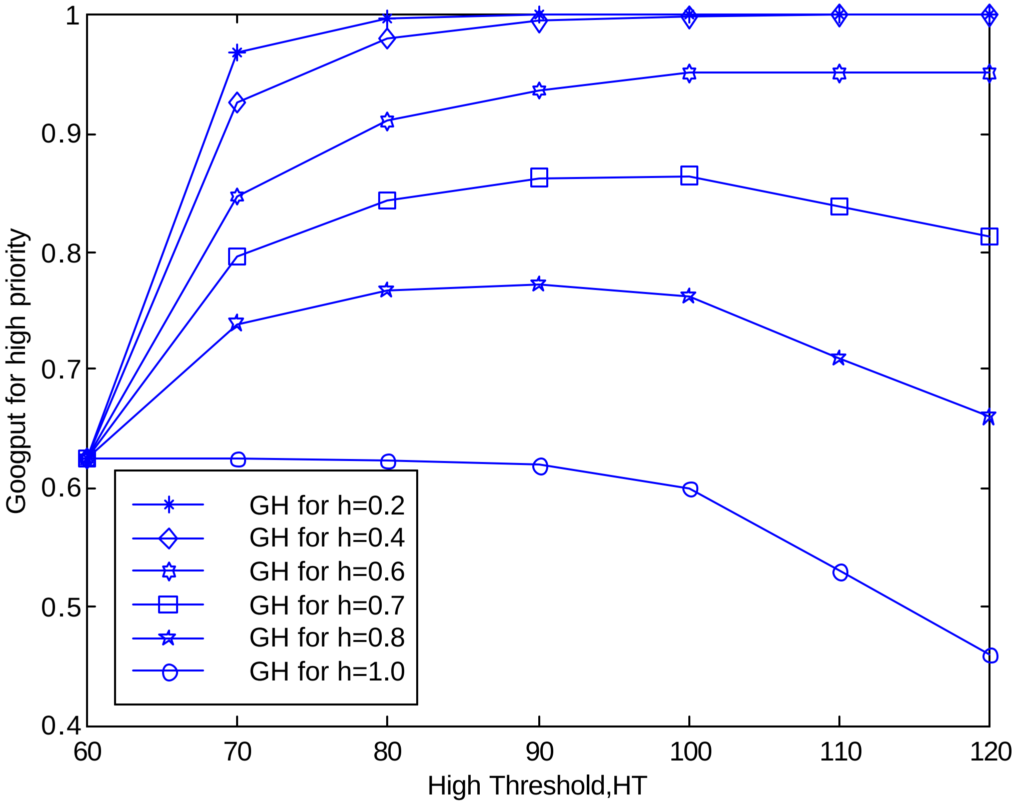
Simulation: N=120, LT=60, HT=80, q=1/6.

Goodput versus load for *h*=0.2, 0.5, 0.8



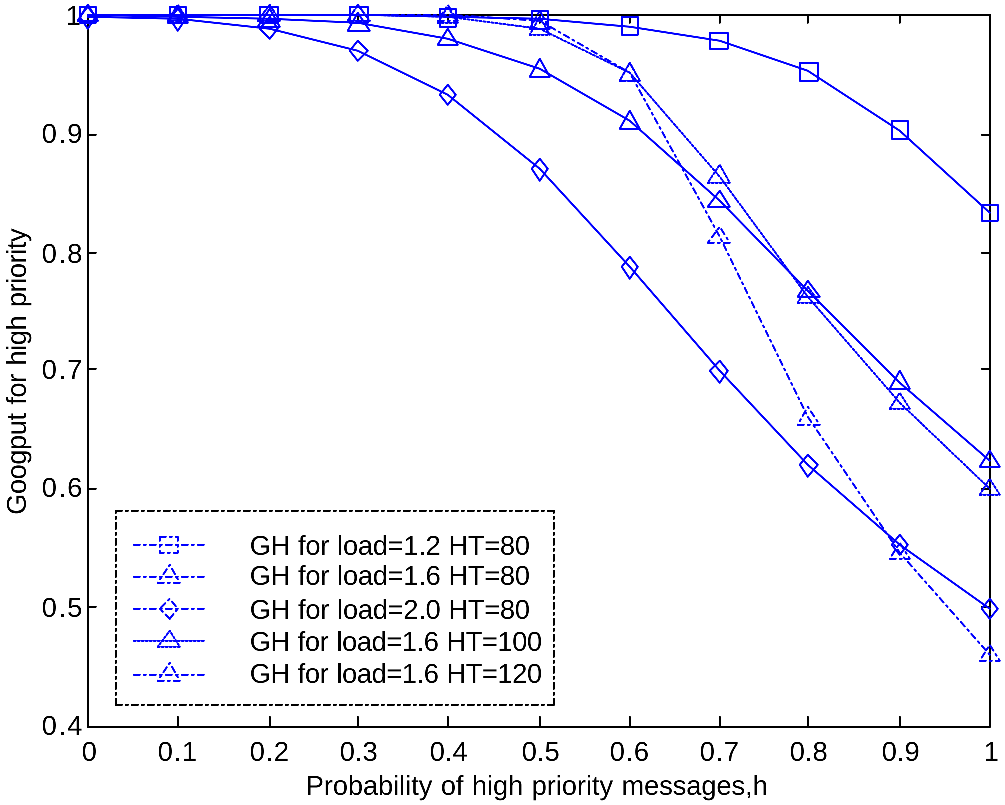
Simulation: N=120, LT=60, HT=80, q=1/6.

Goodput for high priority vs. HT



Simulation: LT=60, *q*=1/6

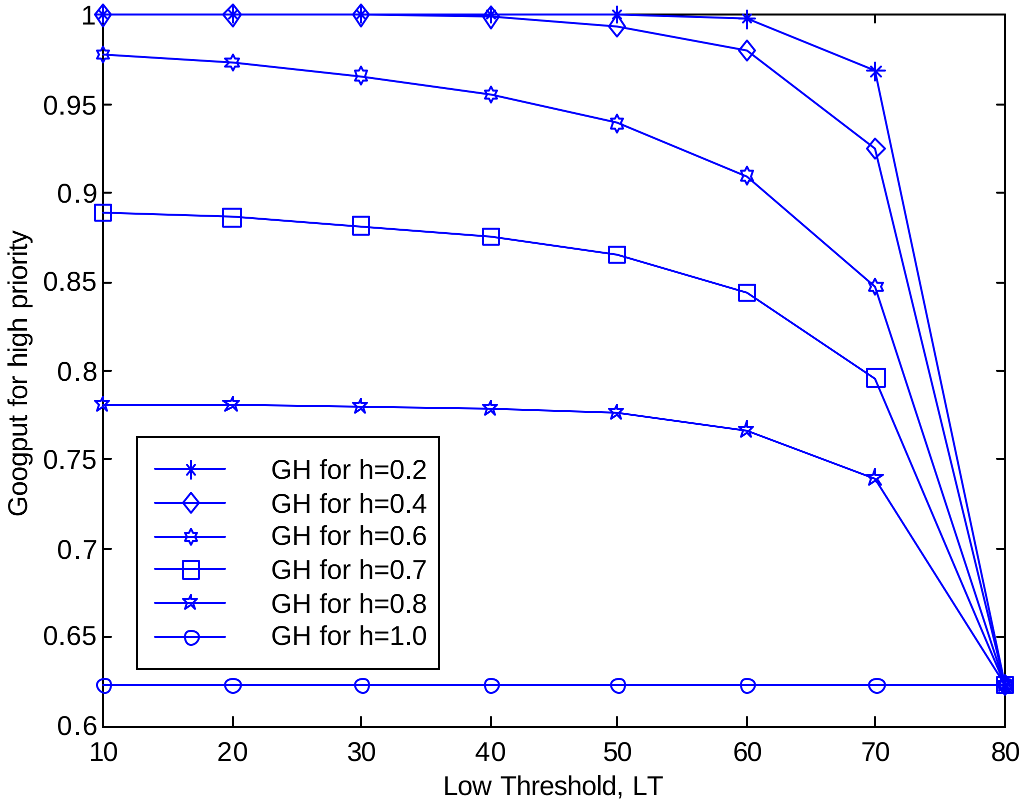
Goodput of high priority vs. *h*



Simulation: *LT*=60

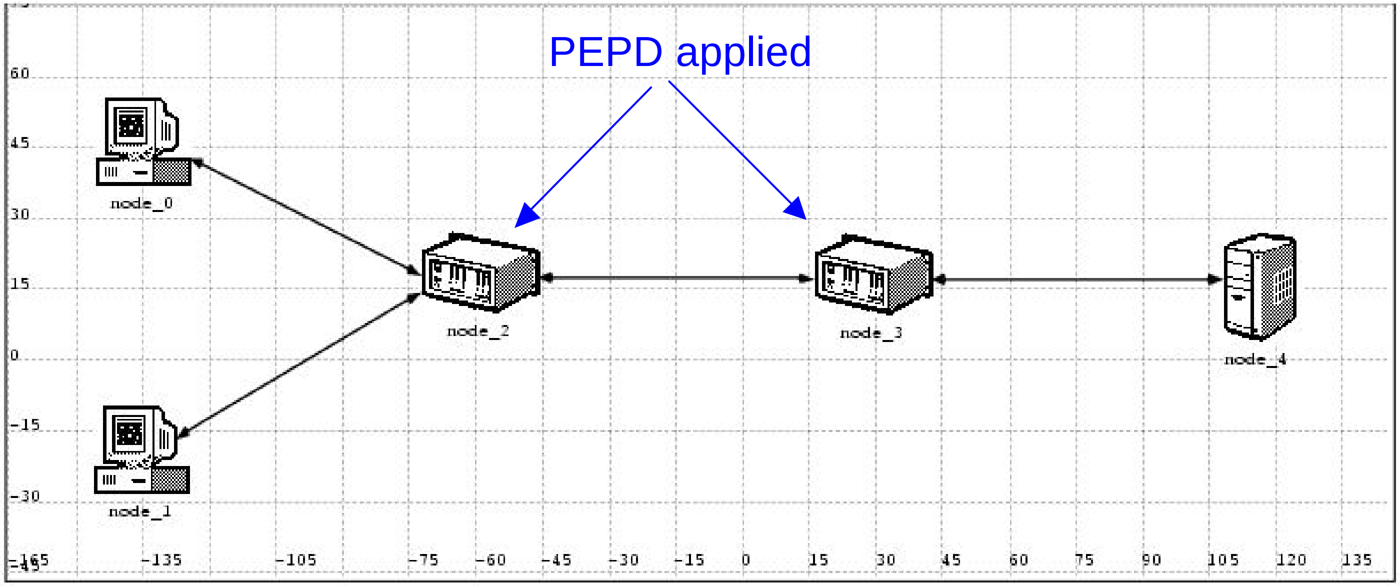
Mohammed Atiquzzaman, University of Dayton,
Email: atiq@ieee.org

Goodput for high priority versus *LT*



Simulation: *N*=120, *HT*=80, load=1.6, *q*=1/6.

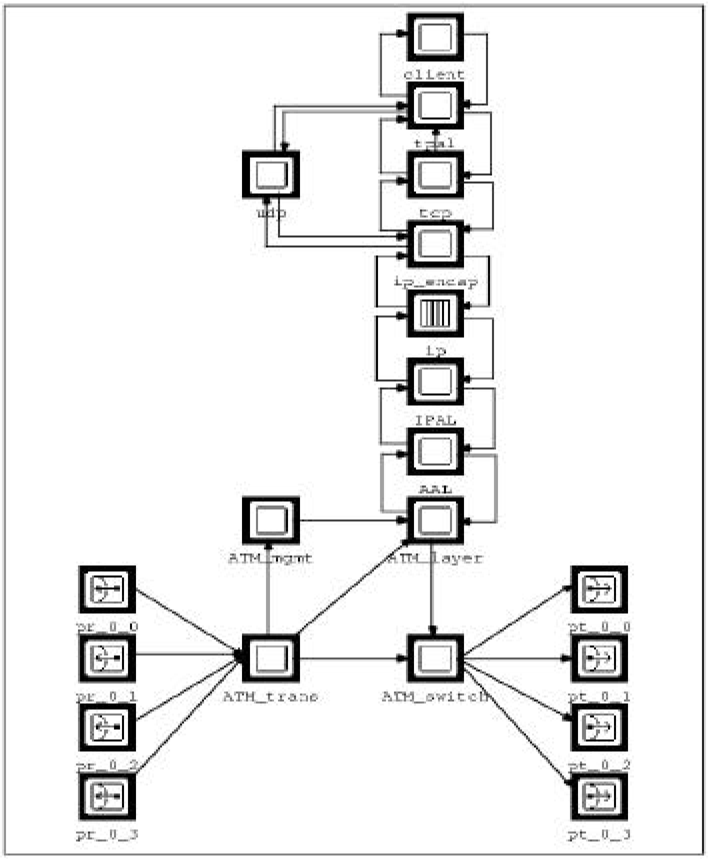
OPNET Simulation Configuration



PEPD applied

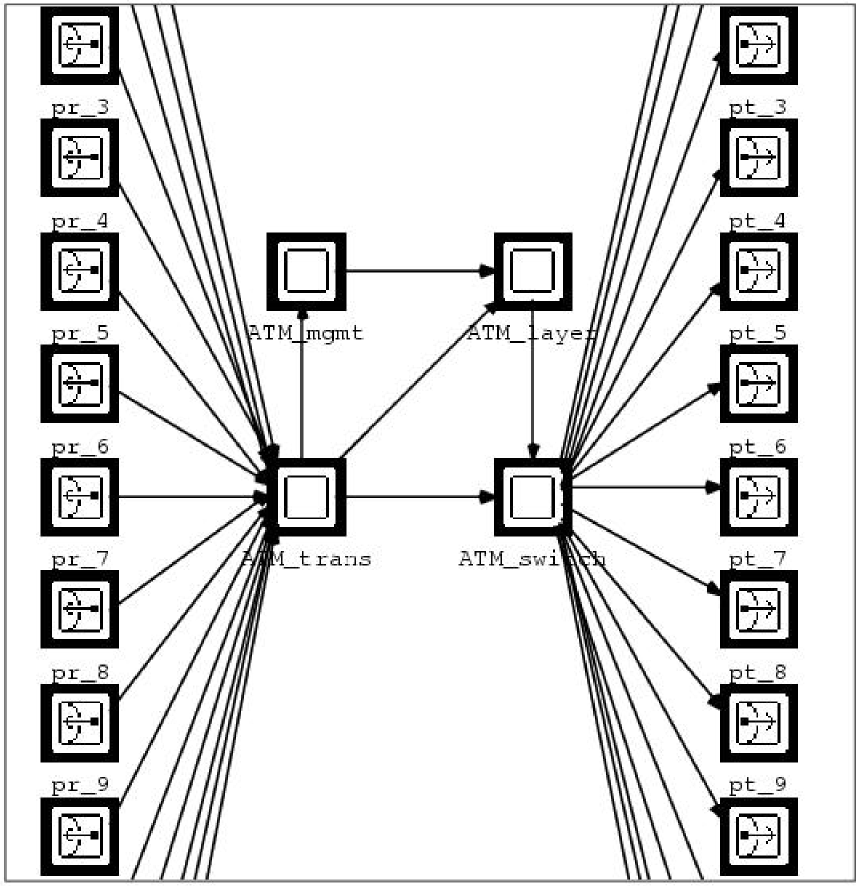
DS-ATM Protocol Stack

- AAL Layer marks the End of Packet.
- ATM_layer changes the CLP bit depending on the packet of the DS service.

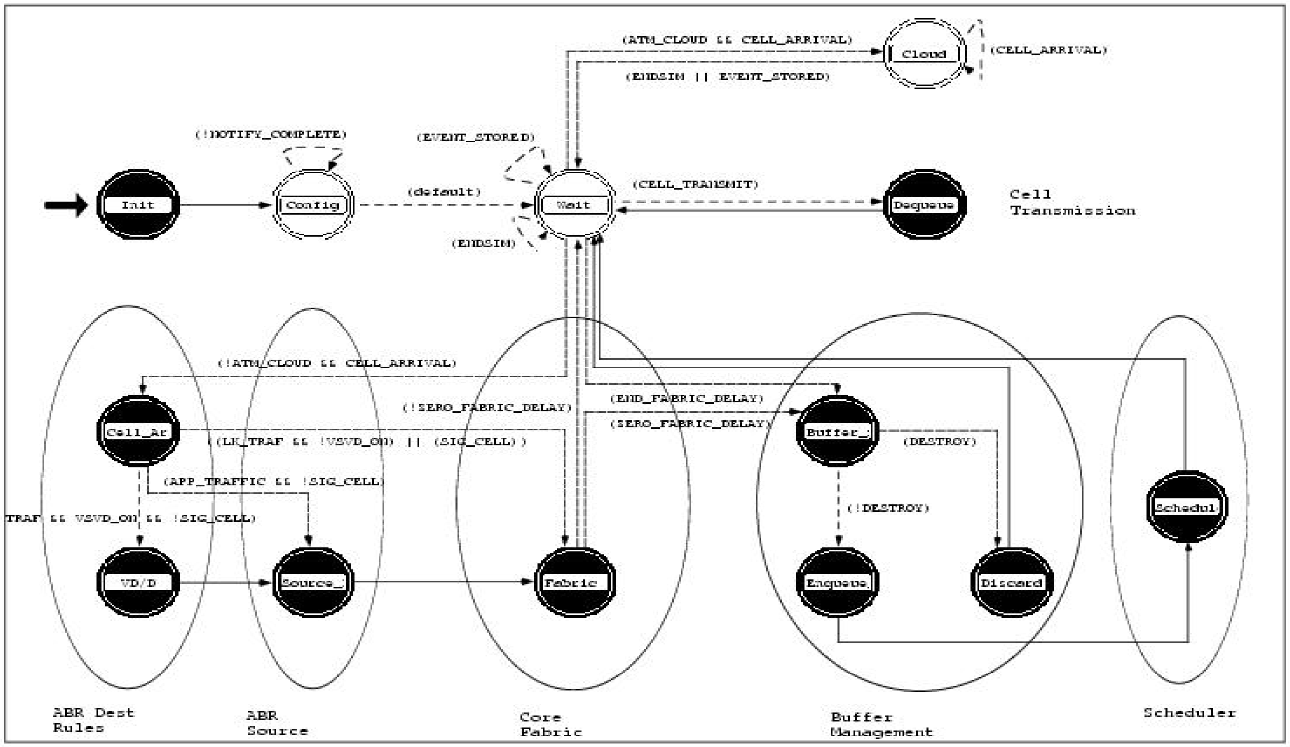


ATM Switch Node

- Support service differentiation in the ATM switch buffer.
- Change the buffer management scheme in the ATM_switch process to Prioritized EPD.



ATM_Switch Process



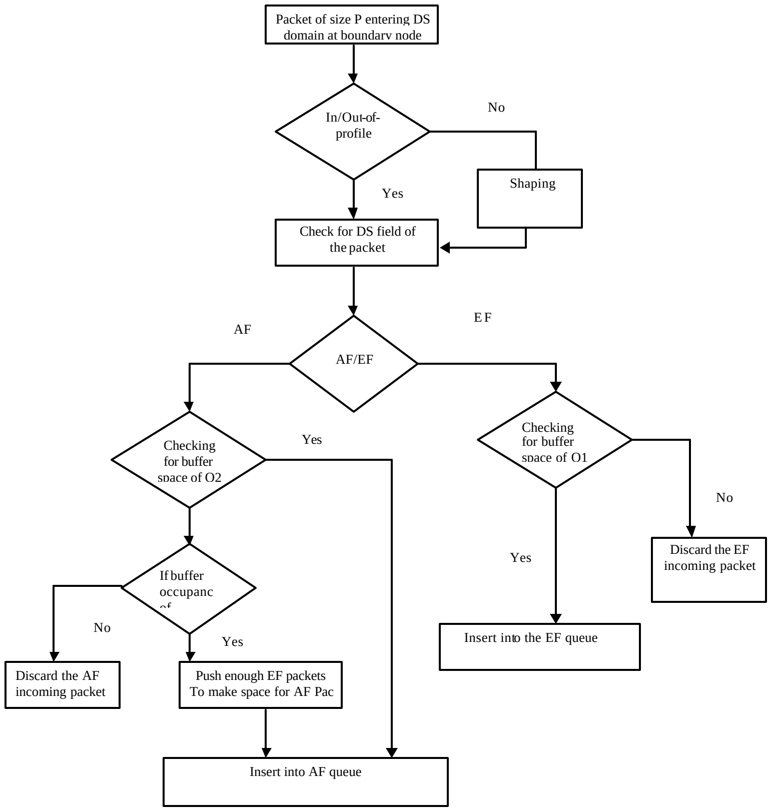
Implements the PEPD buffer management to support service differentiation.

Task 2

Mapping of

IS over DS

Traffic entering DS domain


Packet of size P entering DS domain at boundary node
In/Out-of-profile
No
Shaping
Yes
Check for DS field of the packet
AF
E F
AF/EF
Checking for buffer space of O2
Yes
Checking for buffer space of O1
No
Discard the EF incoming packet
Yes
If buffer occupanc of
No
Yes
Discard the AF incoming packet
Push enough EF packets To make space for AF Pac
Insert into the EF queue
Insert into AF queue

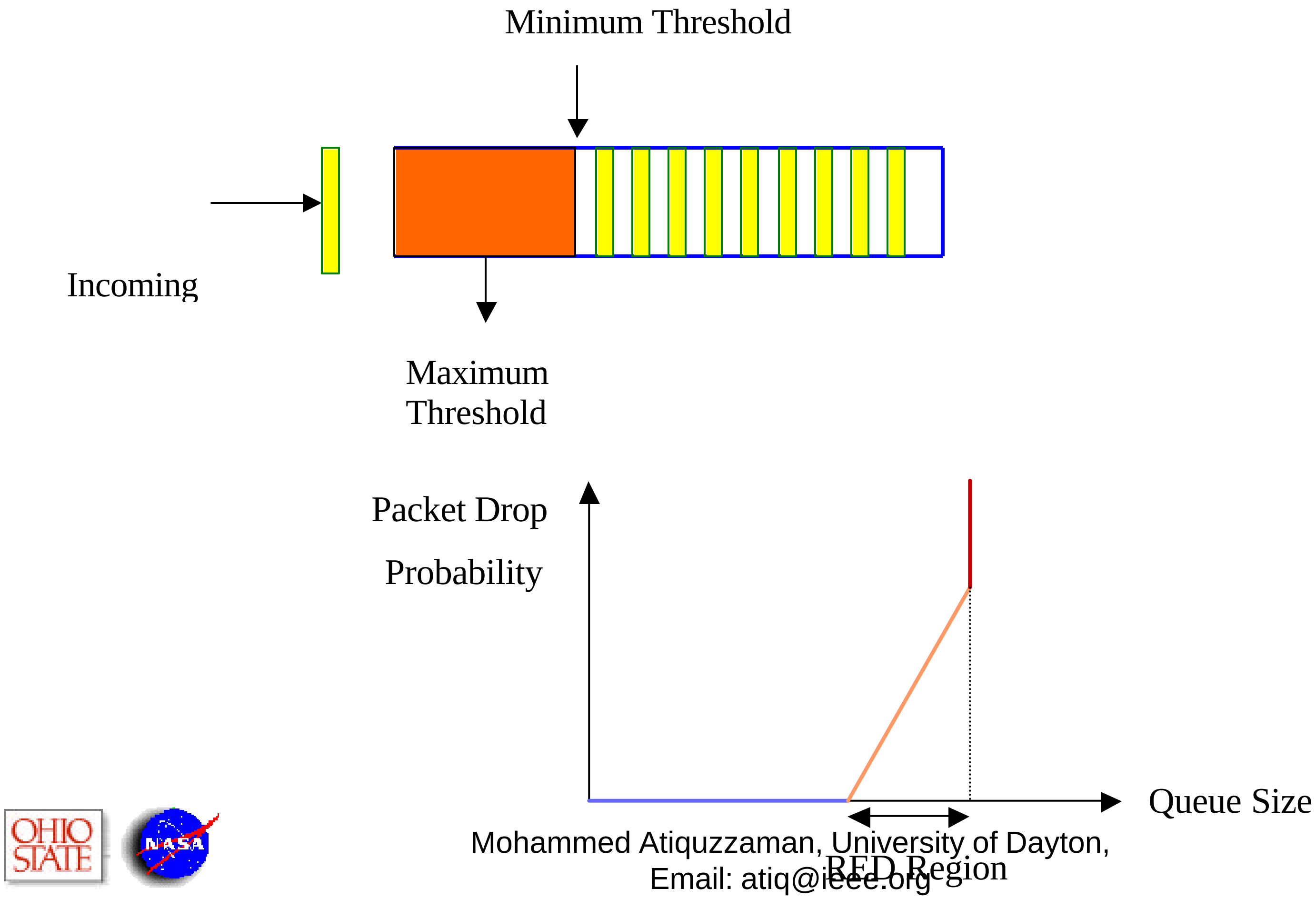
Differentiated Services

- *Classification*: Based on IP header field classifies into BA to receive particular per hop behavior (PHB)
- *Metering*: Measuring the traffic against token bucket to check for resource consumption
- *Shaping*: Treatment of out-of-profile traffic by placing it in a buffer.
- *Dropping*: Non-conformant traffic can be dropped for congestion avoidance
- *Admission Control*: Limiting the amount of traffic according to the resources in the DS domain.
  - Implicit Admission Control: Performed at each router
  - Explicit Admission Control : Dynamic resource allocation by a centralized bandwidth broker

Various PHB's

- Expedited Forwarding(EF PHB)
- Assured Forwarding (AF PHB)
- Best Effort (Default)

Queue Implementation (RED)



QoS Specifications

- **Bandwidth:**
- **Latency:**
- **Jitter:**
- **Loss:**

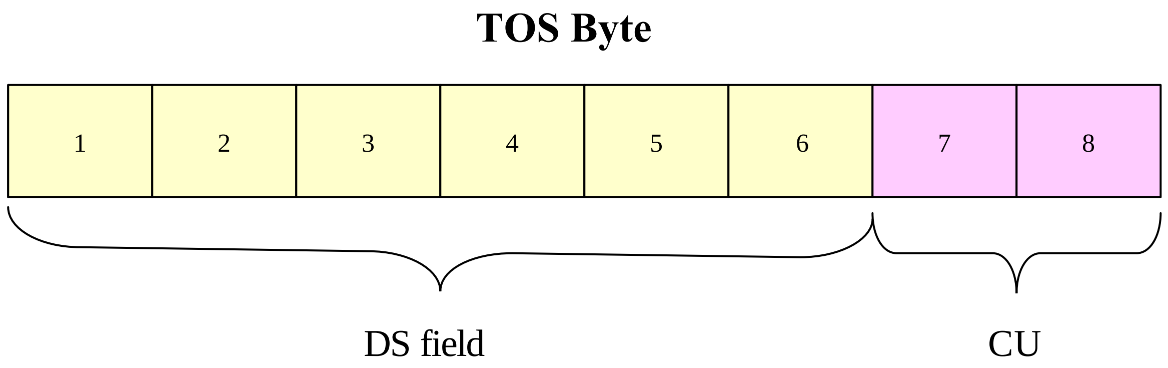
Service Mapping from IS-DS

- Provide different levels of service differentiation.
- Provide QoS to multimedia and multicast applications.
- Scalability in terms of resource allocation.
- There is no over head due to per flow state maintenance at each router.
- Forwarding at each router according to the DSCP code.
- PHB's along the path provide a scheduling result approximating the QoS requirements and results in IS

Integrated Service	Differentiated Service
Guaranteed Load	Expedite Forwarding
Controlled Load	Assured Forwarding
Best effort	Default best effort

DS functionality

- Per Hop behavior (PHB)
- Behavior Aggregate (BA)
- Differentiated Services Code Point (DSCP)



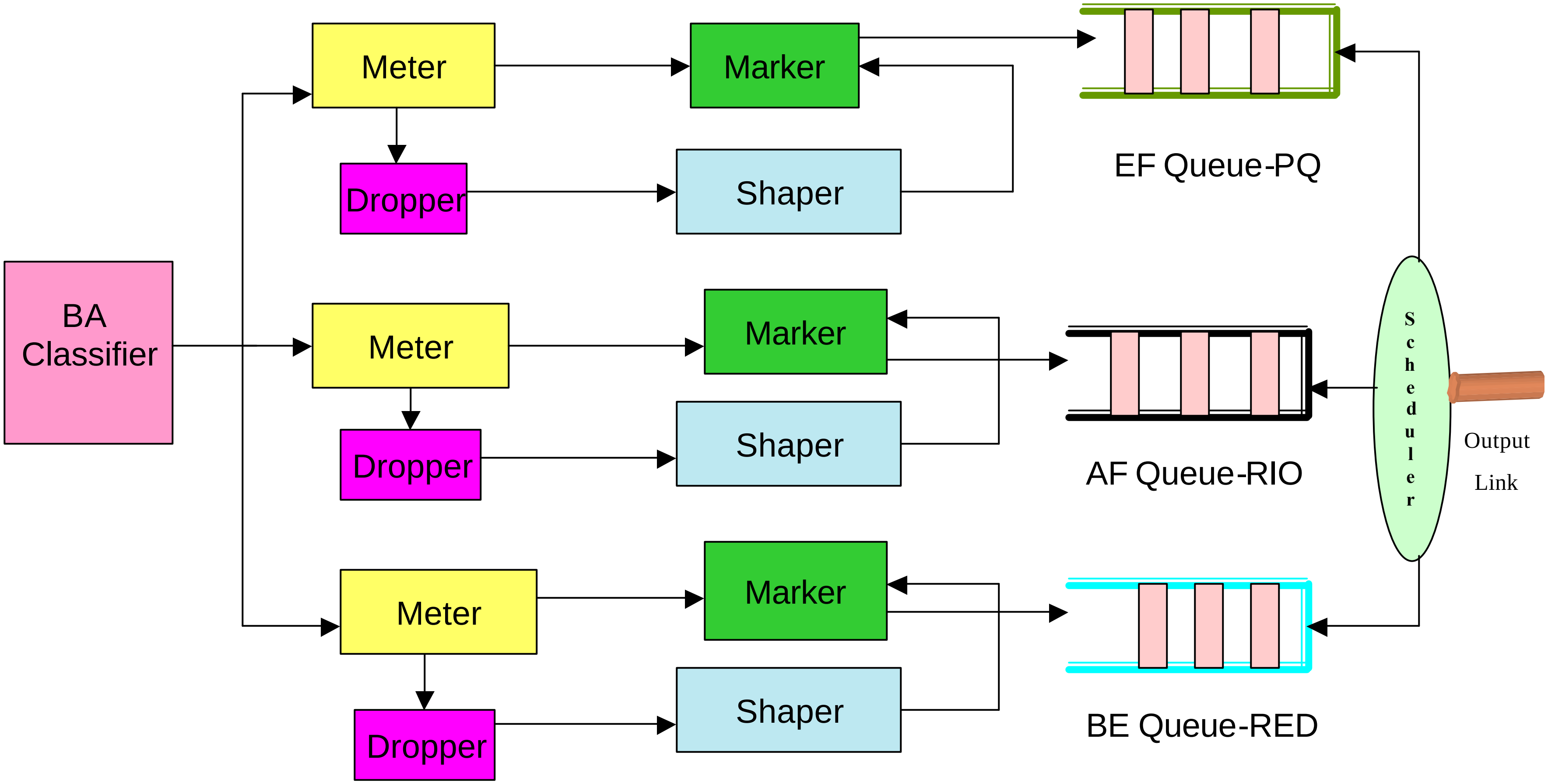
Guaranteed load - EF PHB

- Guaranteed traffic performance can be met effectively using the EF PHB with proper policing and shaping functions.
- Shaping Delay
- Queuing Delay
- Packets in the Scheduler

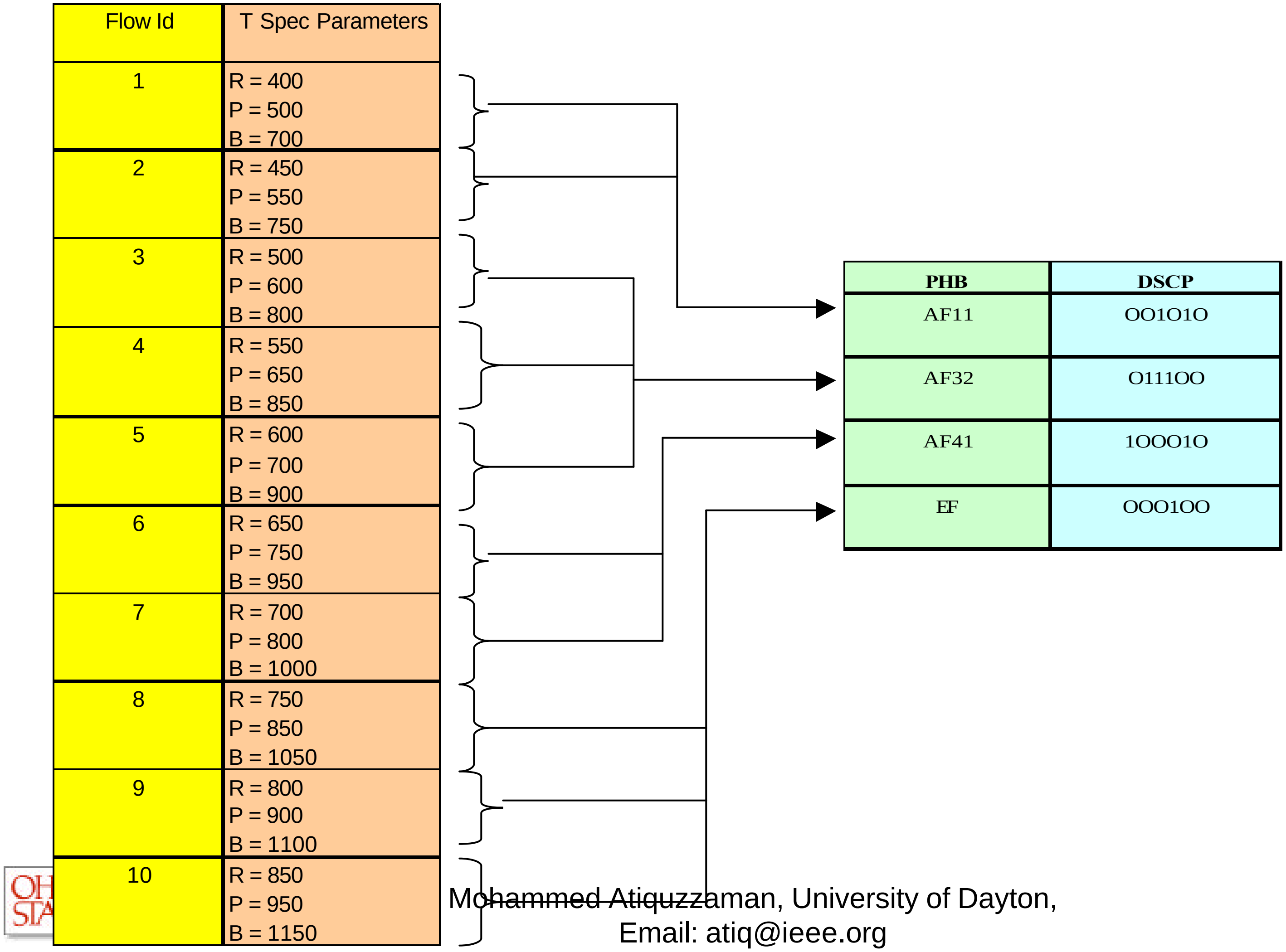
Controlled Load - AF PHB

- Classified into delay classes based on the B/R ratio of Tspec for each delay class; Aggregate Tspec is constructed for all the admitted traffic.
- For each delay class, police the traffic against a token bucket derived above.
- Size of the queue is set to limit the queuing delay of AF requirement.
- RIO dropping parameters are set according to the drop precedence of the AF class.
- AF instance service rate is set to bandwidth sufficient enough to meet the delay and loss requirements of the CL traffic.
- Bandwidth distributed between AF and BE to prevent the BF from starvation.
- Scheduling done with WFQ (Weighted Fair Queuing) or WRR (Weighted Round Robin)

Traffic Conditioning at DS Boundary



Mapping Table for IS-to-DS



Flow Id	T Spec Parameters

1	R = 400
	P = 500
	B = 700

2	R = 450
	P = 550
	B = 750

3	R = 500
	P = 600
	B = 800

4	R = 550
	P = 650
	B = 850

5	R = 600
	P = 700
	B = 900

6	R = 650
	P = 750
	B = 950

7	R = 700
	P = 800
	B = 1000

8	R = 750
	P = 850
	B = 1050

9	R = 800
	P = 900
	B = 1100

10	R = 850
	P = 950
	B = 1150

PHB	DSCP
AF11	001010
AF32	011100
AF41	100010
EF	000100

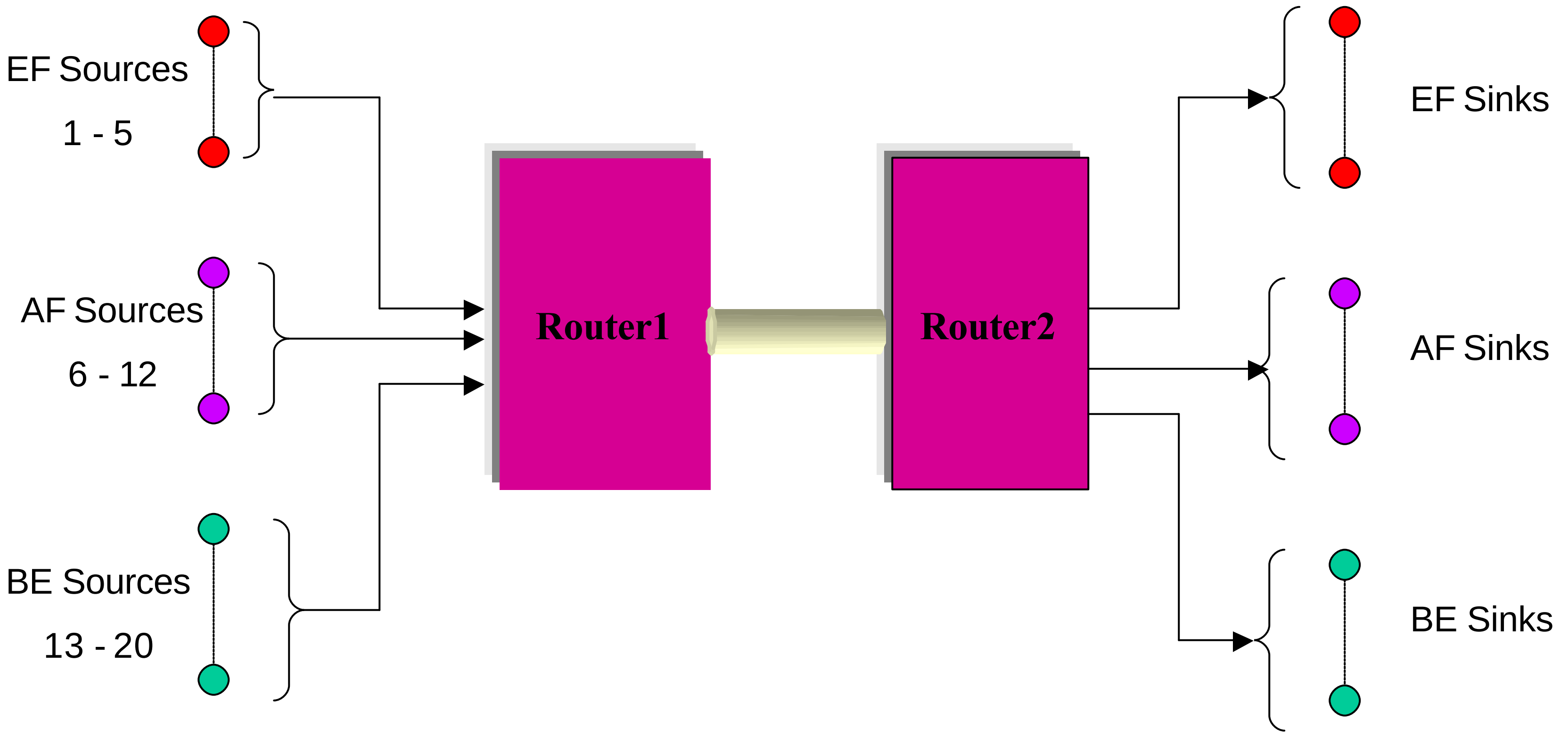
Mapping of IS to DS

- Tspec parameters indicating resource reservation taken from RSVP signaling.
- Table entry contains Tspec parameters, flow IDs, PHB groups and DSCP values.
- Measures actual traffic flow rate against a token bucket according to the initial stored table entry.
- If the traffic is in-profile with the requested reservation, it classifies the packet and marks it with the available DSCP, which can approximately assure the requested QoS.
- The out-of profile traffic is stored in a buffer and shaped to be in conformance with the requested traffic profile.

Mapping of IS to DS (contd.)

- Packets are forwarded in the DS domain according to the DSCP value and the PHB group.
- The forwarding treatment is basically concerned with the queue management policy and the priority of bandwidth allocation; these ensure the required minimum queuing delay, low jitter and maximum throughput.
- Depending on the implementations of the PHB’s inside the network, queue management could be RED, WRED, PQ, WFQ.

IS-DS Simulation Configuration



Task 3

Interoperability with Aeronautical Telecommunications Networks (ATN)

Overview of ATN

- Aeronautical Telecommunications Network.
- Supporting data link based ATC application & AOC.
- Integrating Air/Ground & Ground/Ground data communications network into a global internet serving ATC & AOC.
- Introducing a new paradigm of ATC based on data link rather than voice communications.
- Operating in a different environment with different data communication service provider.
- Supporting the interconnection of Ess & Iss using a variety of subnetwork types.

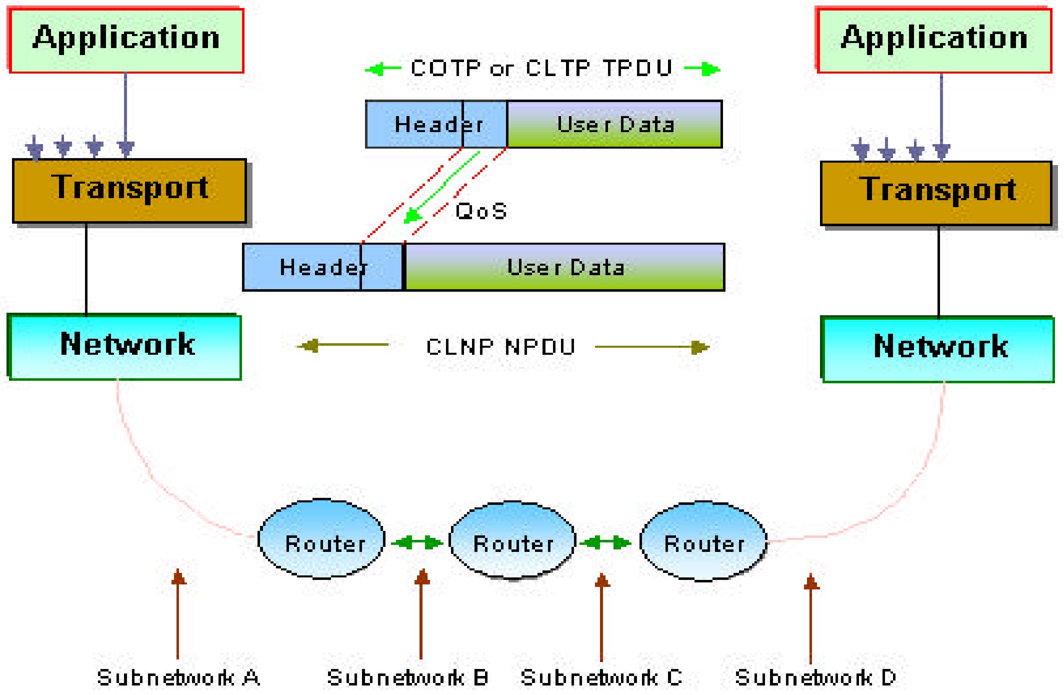
Purpose of ATN

- Using the existing infrastructure.
- High availability.
- Mobile Communications.
- Prioritized end-to-end resource management.
- Scalability.
- Policy based routing.
- Future proofing

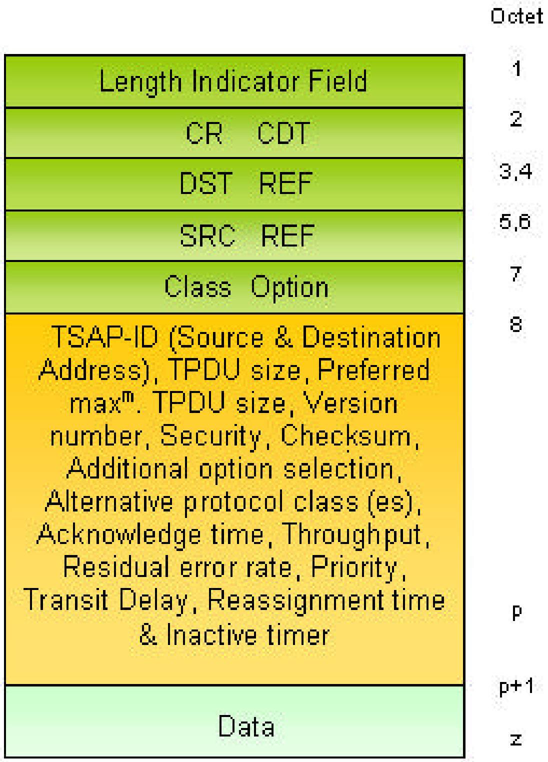
QoS of ATN

- Priority
- Transit Delay
- Error Probability
- Cost
- Security
- Reliability

Model of Transport Layer



Structure of TPDU

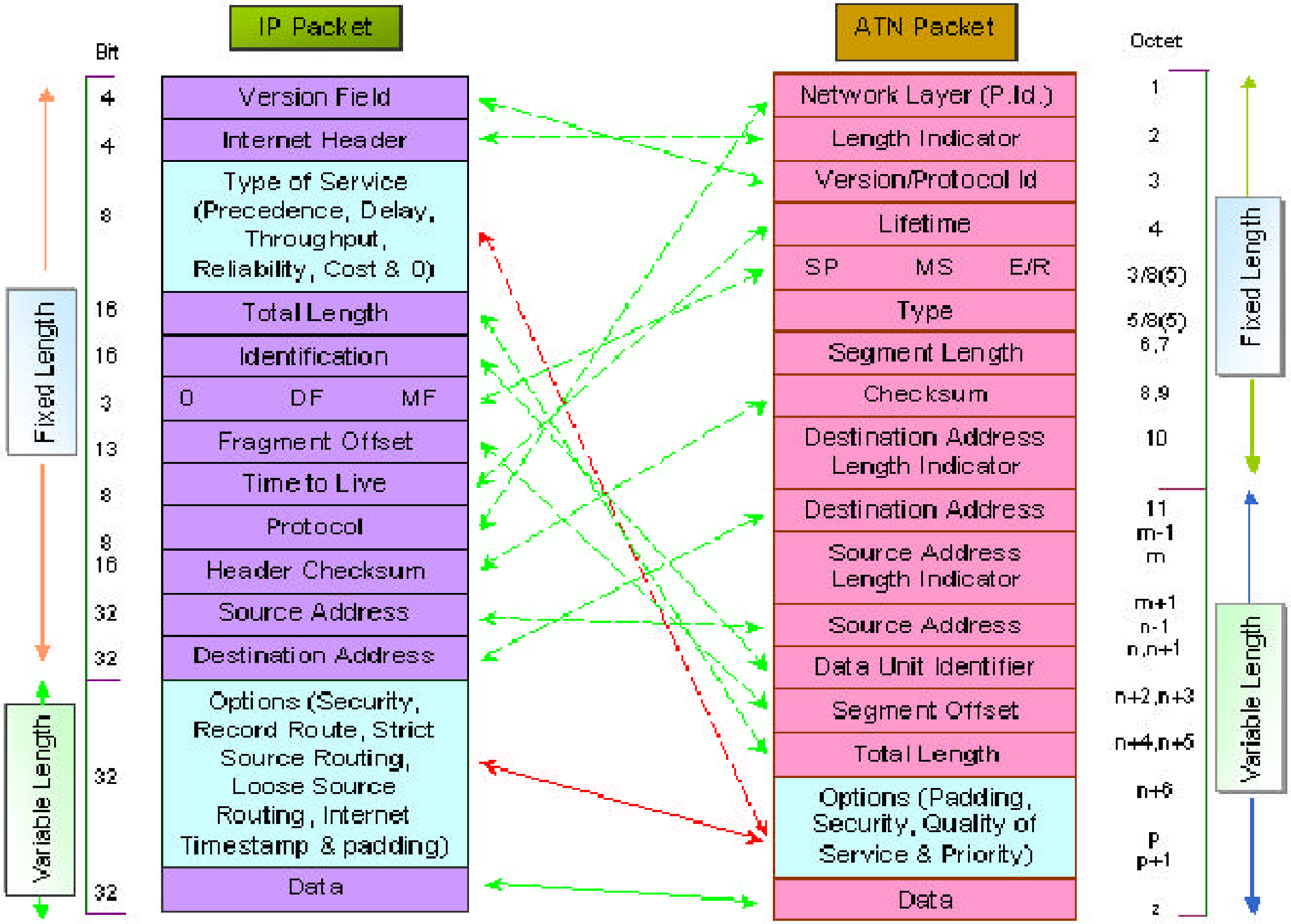


Field	Octet
Length Indicator Field	1
CR CDT	2
DST REF	3,4
SRC REF	5,6
Class Option	7
TSAP-ID (Source & Destination Address), TPDU size, Preferred max^{m}. TPDU size, Version number, Security, Checksum, Additional option selection, Alternative protocol class (es), Acknowledge time, Throughput, Residual error rate, Priority, Transit Delay, Reassignment time & Inactive timer	8 p
Data	p+1 z

Variable Fields of TPDU header

	No. of Octets
TSAP-ID (Source & Destination Address)	Variable
TPDU size	1
Preferred max^{m}. TPDU size	up to 4
Version number	1
Security	variable
Checksum	
Additional option selection	1
Alternative protocol class(es)	variable
Acknowledgement time	2
Throughput	12 Or 24
Residual error rate	3
Priority	2
Transit Delay	8
Reassignment time	2
Inactivity timer	4
Data	variable

Comparison of ATN & IP Packets



IP Packet

Bit

4 Version Field

4 Internet Header

8 Type of Service (Precedence, Delay, Throughput, Reliability, Cost & 0)

16 Total Length

16 Identification

3 0 DF MF

13 Fragment Offset

8 Time to Live

8 Protocol

16 Header Checksum

32 Source Address

32 Destination Address

32 Options (Security, Record Route, Strict Source Routing, Loose Source Routing, Internet Timestamp & padding)

32 Data

Fixed Length

Variable Length

ATN Packet

Octet

Network Layer (P.Id.) 1

Length Indicator 2

Version/Protocol Id 3

Lifetime 4

SP MS E/R 3/8(5)

Type 5/8(5)

Segment Length 6,7

Checksum 8,9

Destination Address Length Indicator 10

Destination Address 11 m-1

Source Address Length Indicator m

Source Address m+1 n-1

Data Unit Identifier n,n+1

Segment Offset n+2,n+3

Total Length n+4,n+5

Options (Padding, Security, Quality of Service & Priority) n+6 p

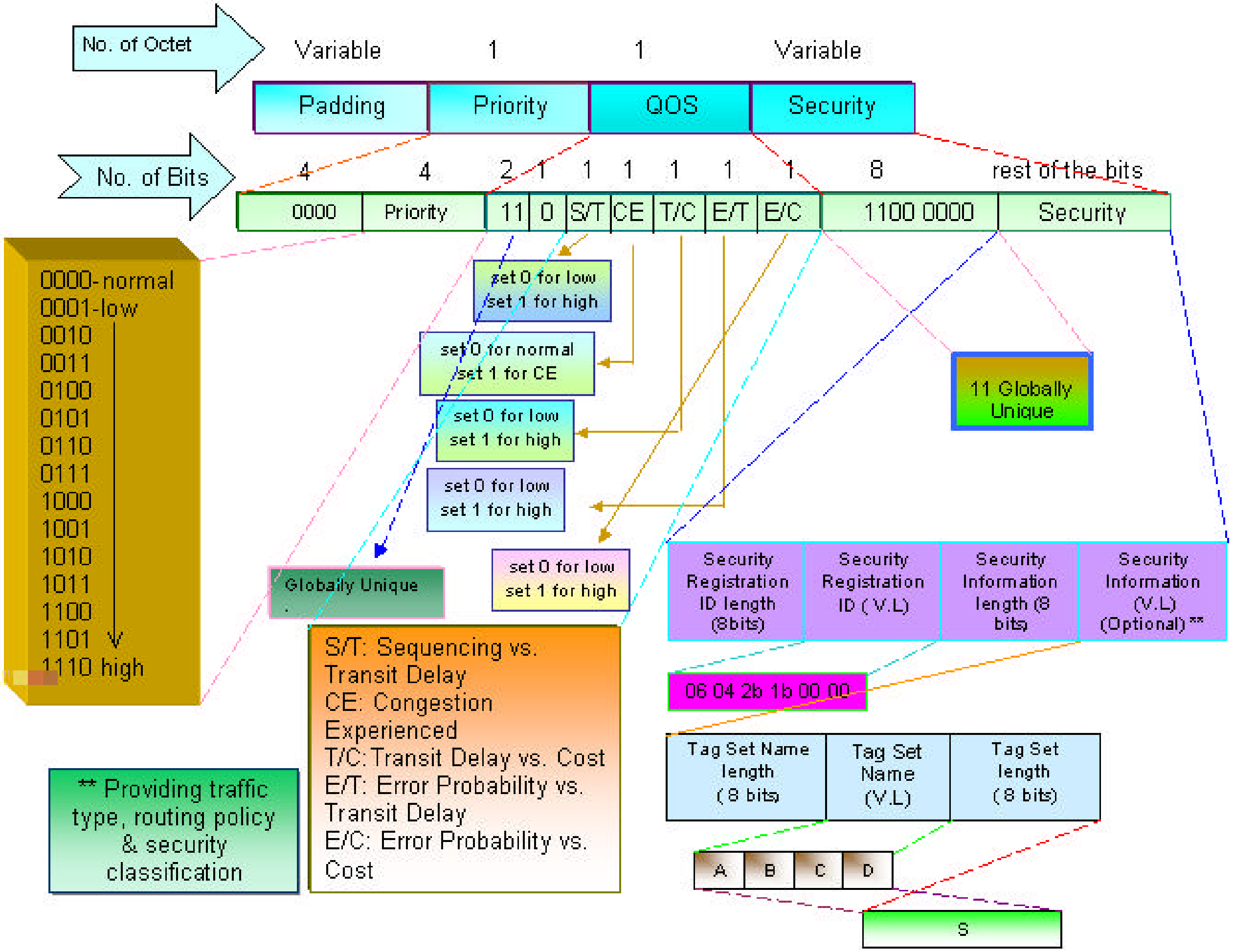
Data p+1 z

Fixed Length

Variable Length

Email: atiq@ieee.org

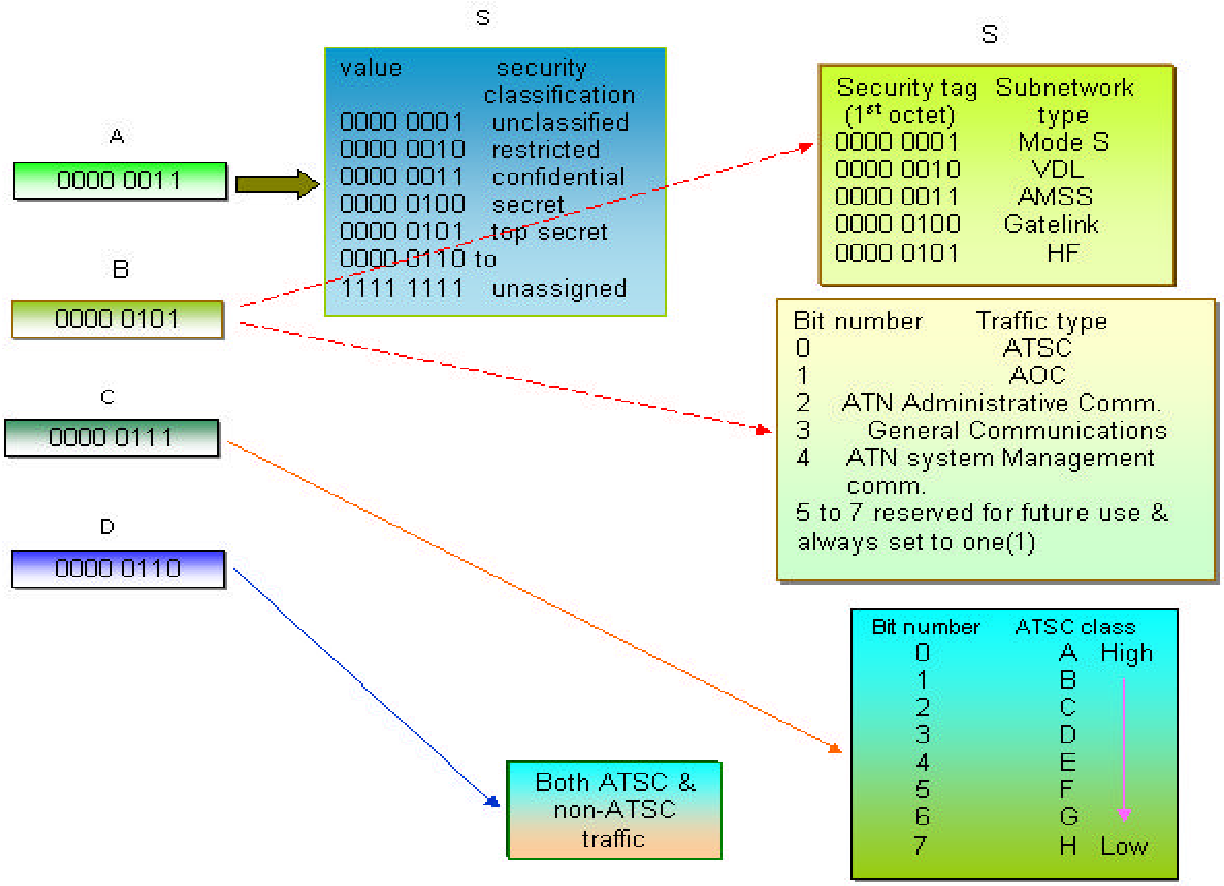
Options Field of ATN Packet



S/T: Sequencing vs. Transit Delay
CE: Congestion Experienced
T/C: Transit Delay vs. Cost
E/T: Error Probability vs. Transit Delay
E/C: Error Probability vs. Cost

** Providing traffic type, routing policy & security classification

Options Field of ATN (contd.)



A

0000 0011

S

value	security classification
0000 0001	unclassified
0000 0010	restricted
0000 0011	confidential
0000 0100	secret
0000 0101	top secret
0000 0110 to 1111 1111	unassigned

B

0000 0101

S

Security tag (1st octet)	Subnetwork type
0000 0001	Mode S
0000 0010	VDL
0000 0011	AMSS
0000 0100	Gatelink
0000 0101	HF

Bit number	Traffic type
0	ATSC
1	AOC
2	ATN Administrative Comm.
3	General Communications
4	ATN system Management comm.

5 to 7 reserved for future use & always set to one(1)

C

0000 0111

Bit number	ATSC class	
0	A	High
1	B	
2	C	
3	D	
4	E	
5	F	
6	G	
7	H	Low

D

0000 0110

Both ATSC & non-ATSC traffic

TPDU & NPDU Priority Translation

Message Catagories	Transport layer Priority	Network layer Priority
Network/System Management	0	14
Distress Communications	1	13
Urgent Communications	2	12
High priority Flight safety Message	3	11
Normal priority Flight safety Message	4	10
Meteorological Communications	5	9
Flight Regularity Communications	6	8
Aeronautical Information Service Message	7	7
Network/System Administration	8	6
Aeronautical Administrative Messages	9	5
<unassigned>	10	4
Urgent Priority Administration & U.N. Charter Communications	11	3
High Priority Administrative & State/Government Communications	12	2
Normal Priority Administrative	13	1
Low Priority Administrative	14	0

Priorities above the bold line are for the communications related to safety & regularity of flight

Conclusions

- Tasks are progressing well and as planned.
- Modeling of Prioritized EPD has been completed.
- OPNET simulation of Prioritized EPD to be continued.
- ns simulation of IS over DS to be continued.
- ATN over DS mapping to be started.

QoS for Real Time Applications over Next Generation Data Networks: Project Status Report Part II

Raj Jain
The Ohio State University
Columbus, OH 43210
jain@cis.ohio-state.edu

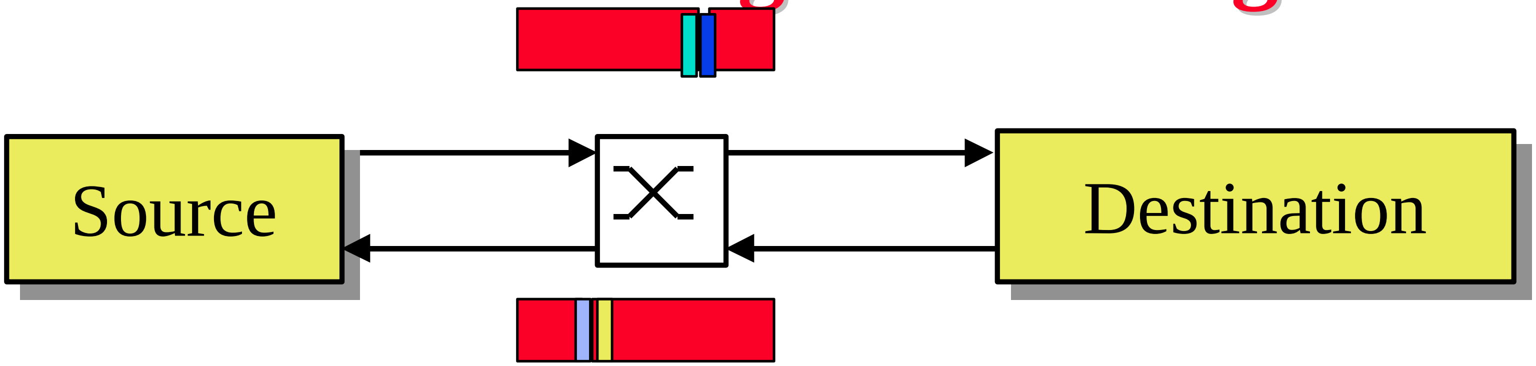
These slides are available at

http://www.cis.ohio-state.edu/~jain/talks/nasa_at0.htm

Overview

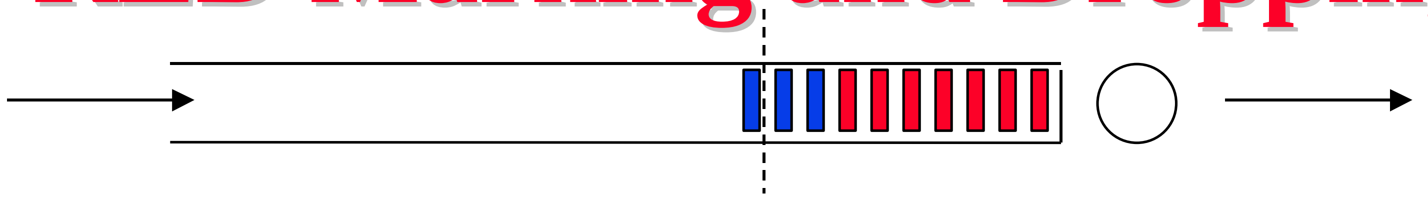
- Task 4: Explicit congestion notification - improvements
- Task 5: Circuit emulation using MPLS
- Acknowledgement: The research reported here was conducted by Chun Lei Liu, Wei Sun, and Dr. Arian Durresi.

Task 4: Buffer Management using ECN

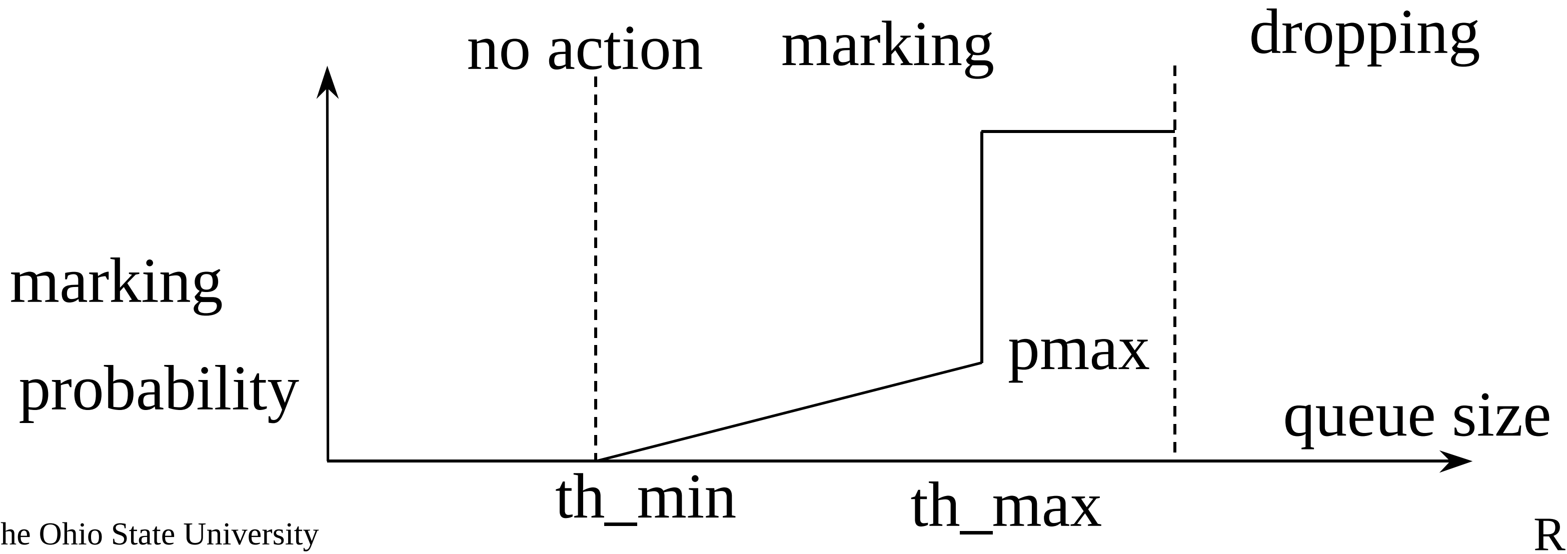


- Explicit Congestion Notification
- Standardized in RFC 2481, expected to be widely implemented soon.
- Two bits in IP header: Congestion Experienced, ECN Capable Transport
- Two bits in TCP header: ECN Echo, Congestion Window Reduced

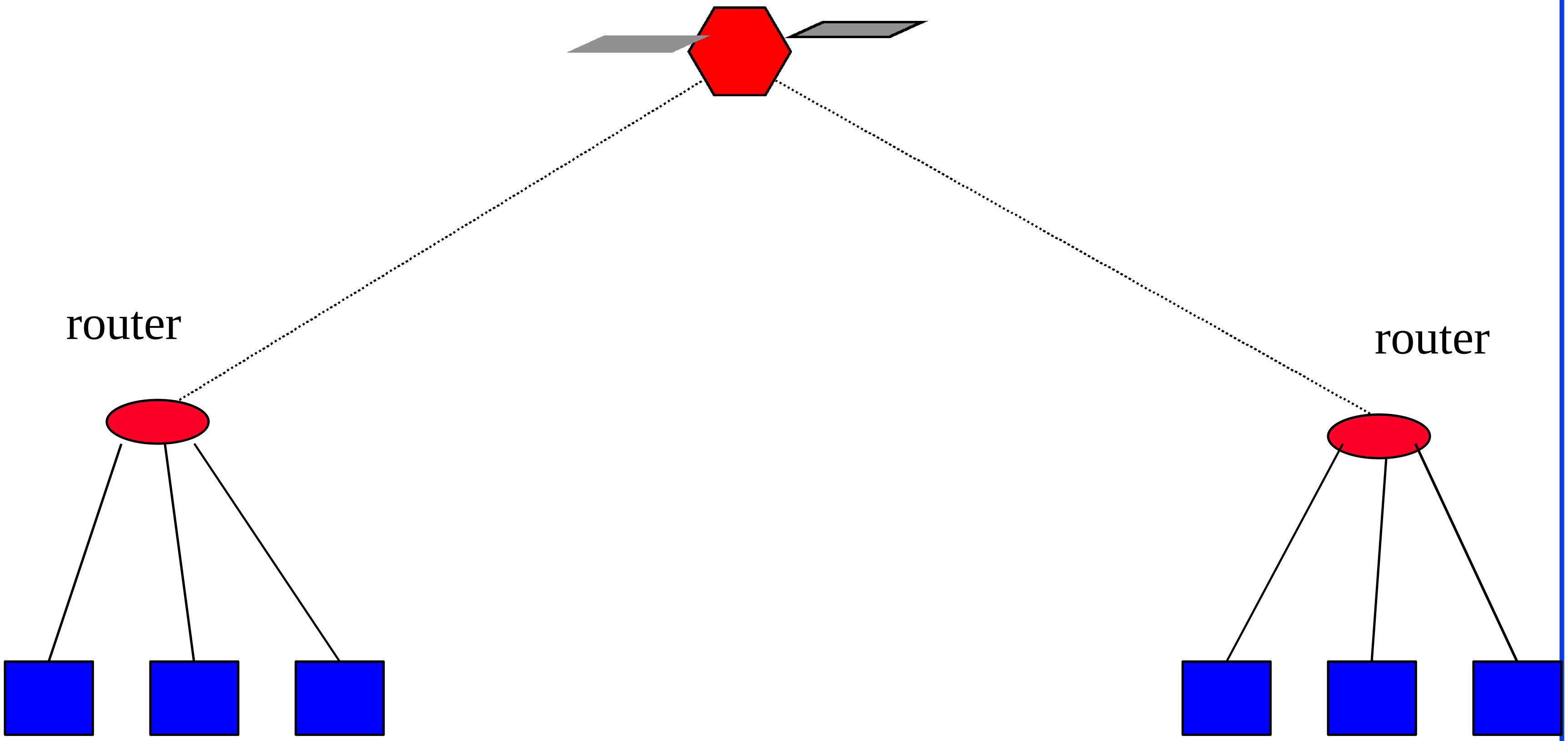
RED Marking and Dropping



- No standard action specified
- One possibility is to randomly mark at lower congestion levels. Random Early Detection (RED) marking instead of dropping.



Simulation Model

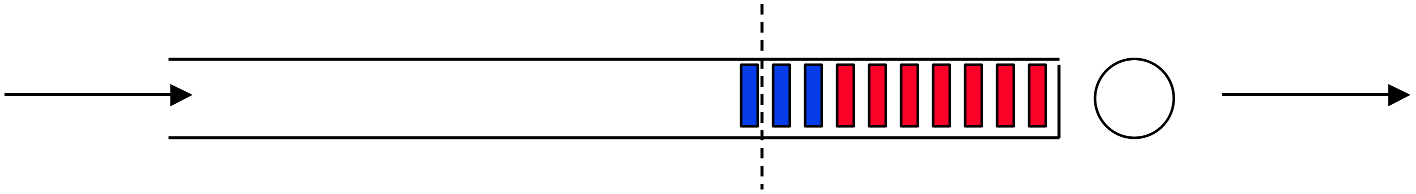


TCP senders

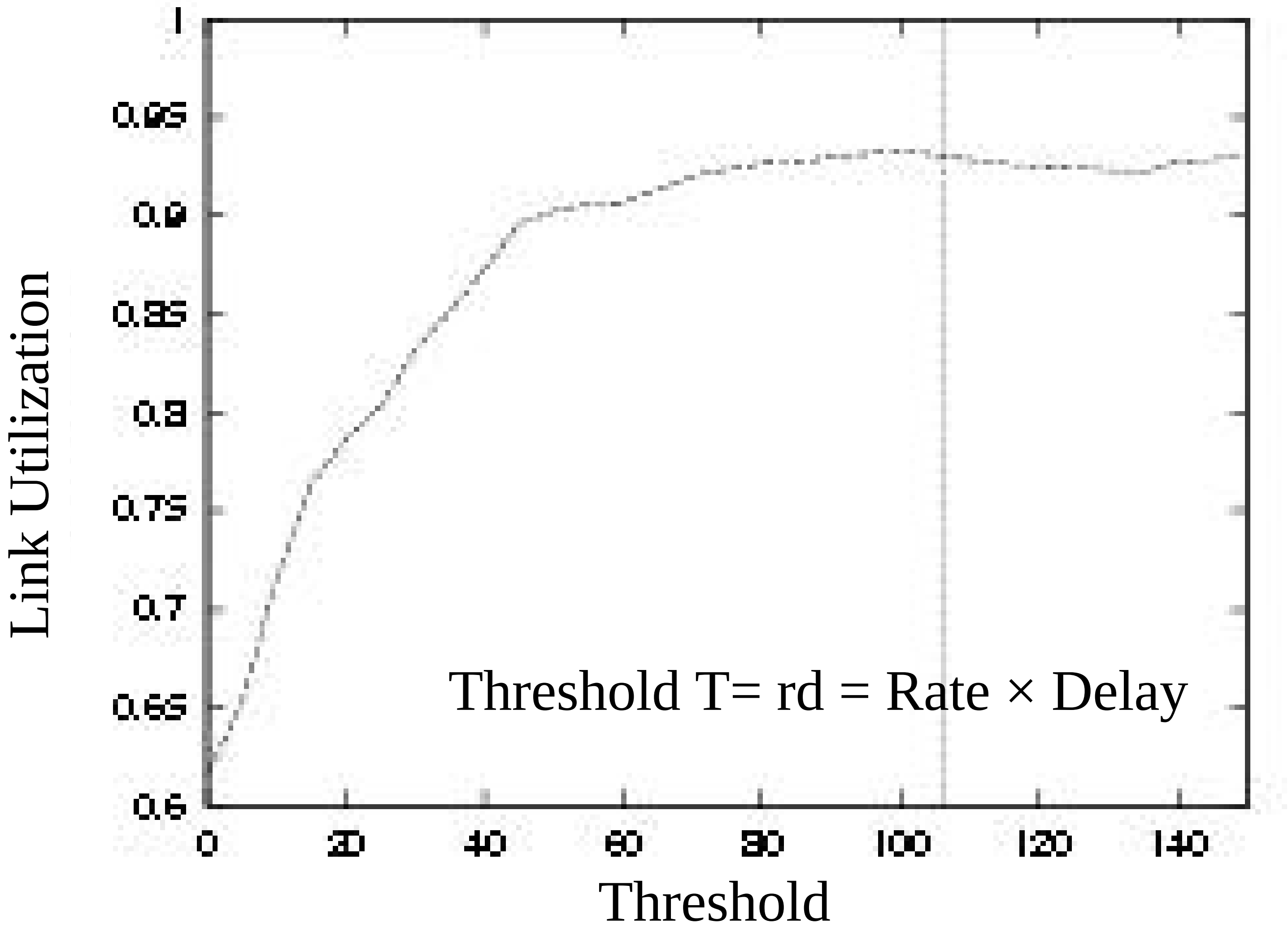
TCP receivers

Threshold & Buffer Requirement

- In order to achieve full link utilization and zero packet loss, ECN routers should
  - have a buffer size of three times the bandwidth delay product;
  - set the threshold as 1/3 of the buffer size;
- Any smaller buffer size will result in packet loss.
- Any smaller threshold will result in link idling.
- Any larger threshold will result in unnecessary delay.

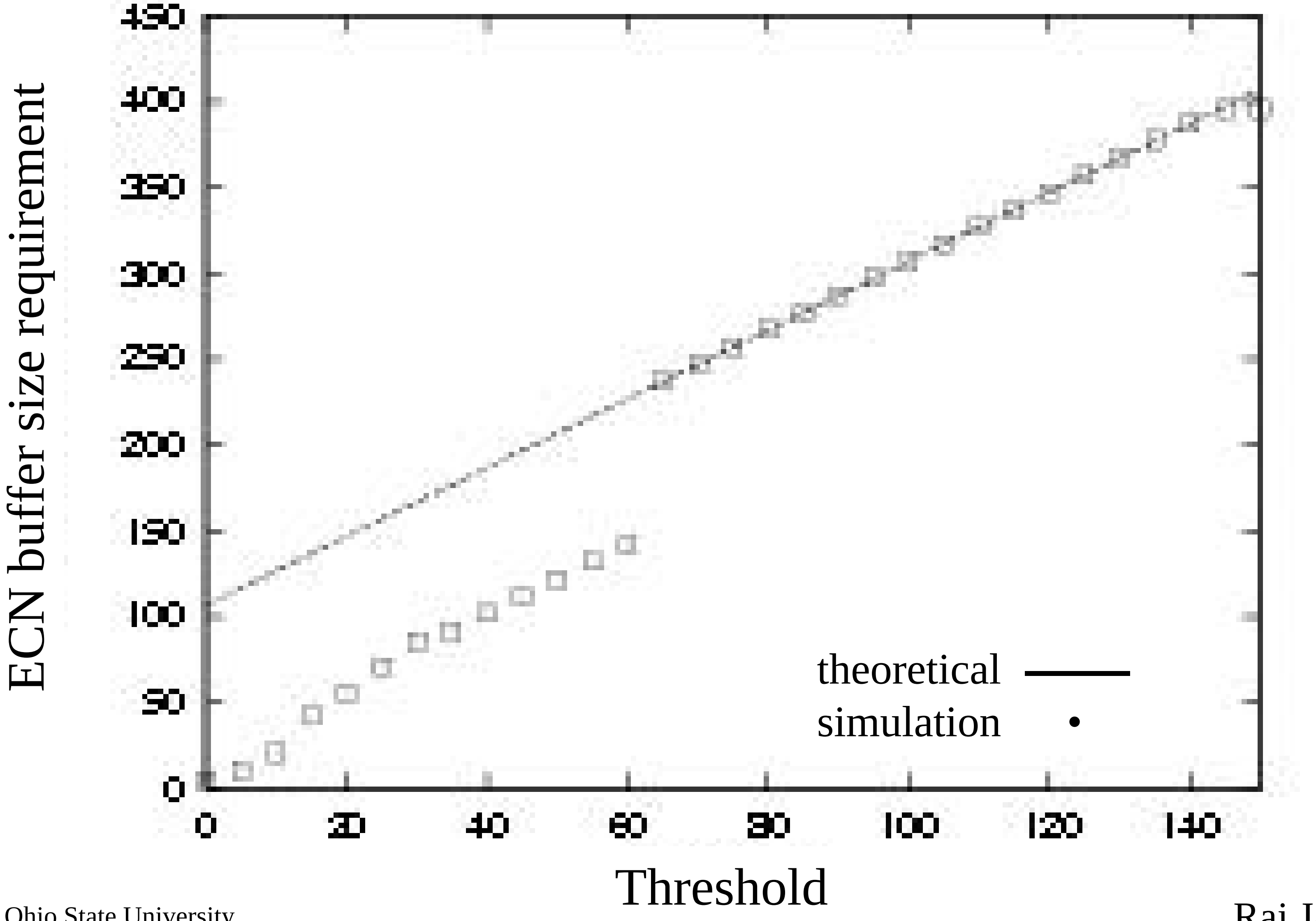


Setting the Threshold

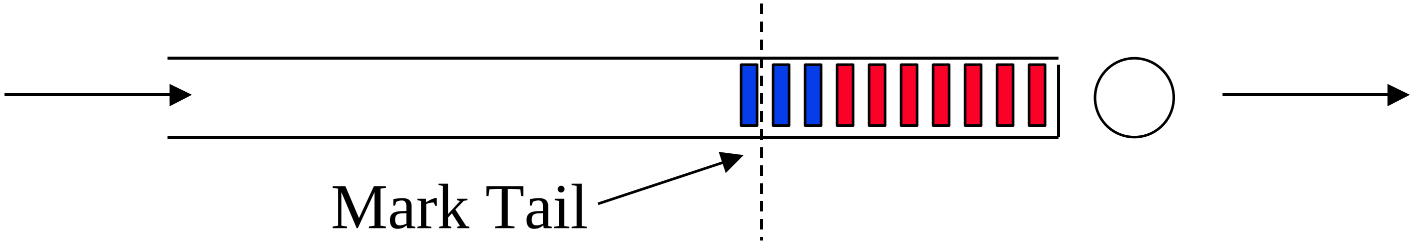


Threshold T= rd = Rate × Delay

Buffer Requirement for No Loss

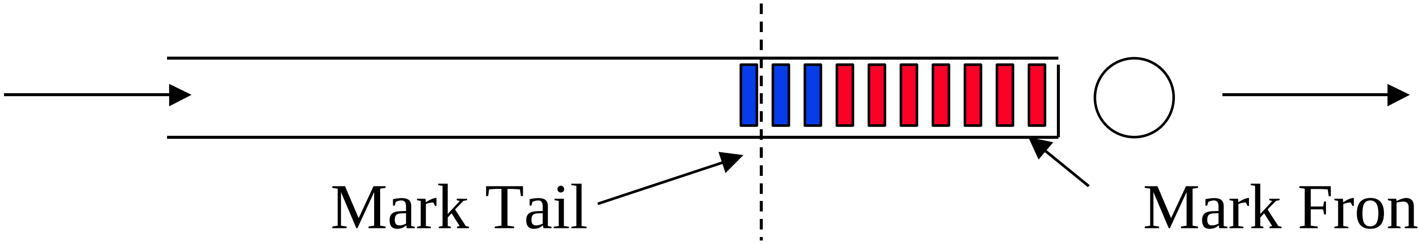


Problem with Mark Tail



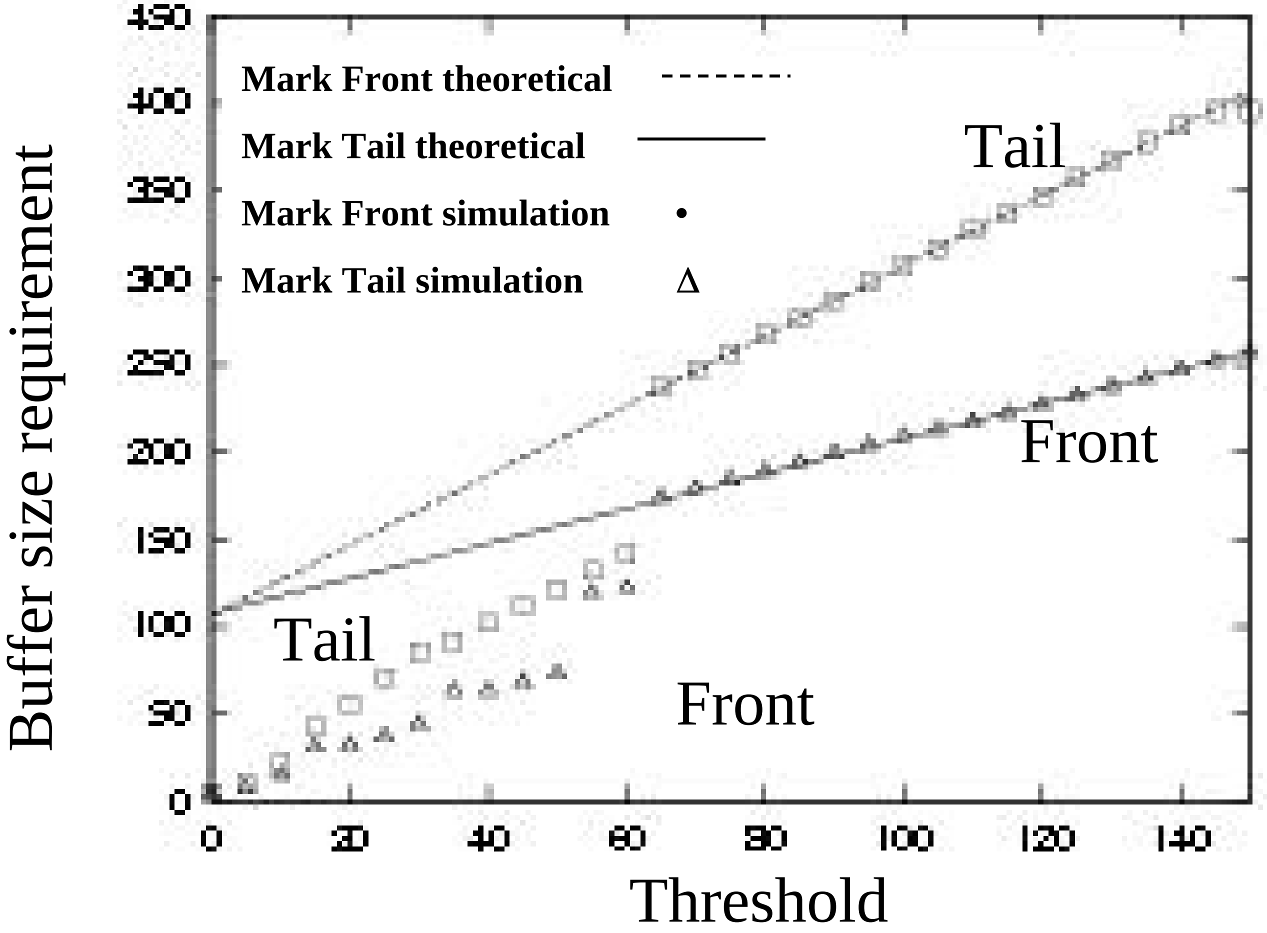
- Current ECN marks packets at the tail of the queue:
  - Congestion signals suffer the same delay as data;
  - When congestion is more severe, the ECN signal is less useful;
  - Flows arriving when buffers are empty may get more throughput ⇒ unfairness.

Proposal: Mark Front



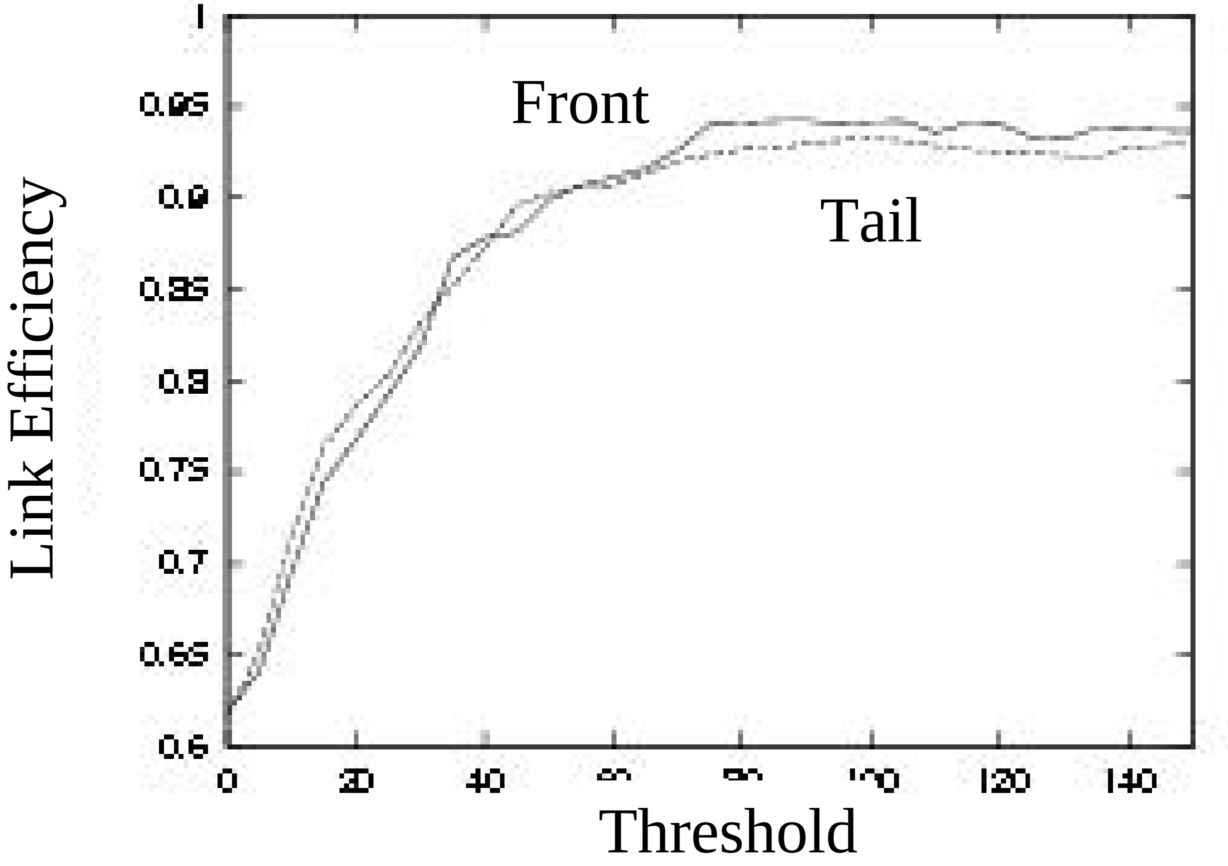
- Simulation Analysis has shown that mark-front strategy
  - Reduces the buffer size requirement,
  - Increases link efficiency,
  - Avoids lock-in phenomenon and improves fairness,
  - Alleviates TCP's discrimination again large RTTs.

Buffer Requirement



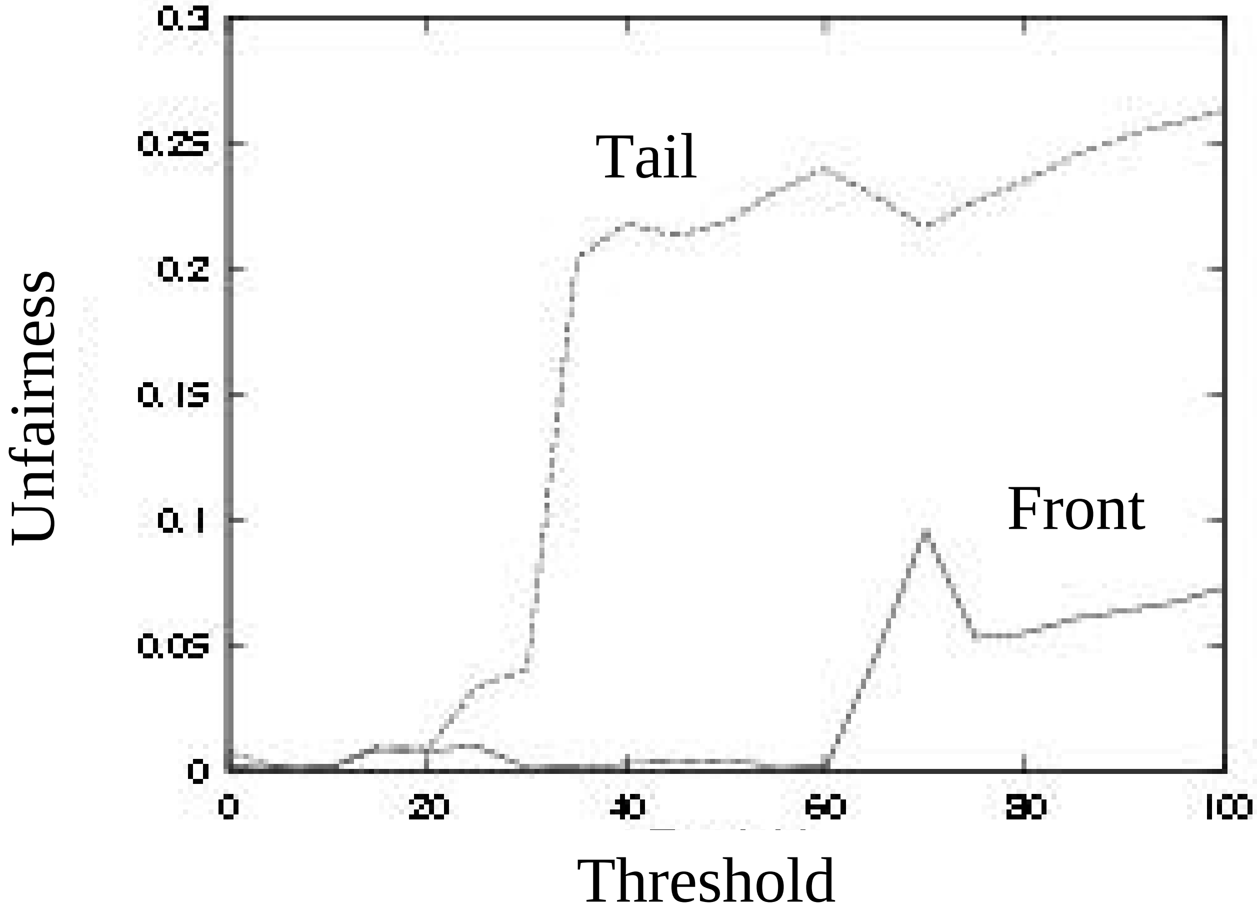
- ❑ Our theoretical bound is tight when threshold ≥ rd
- ❑ Mark Front requires less buffer

Link Efficiency



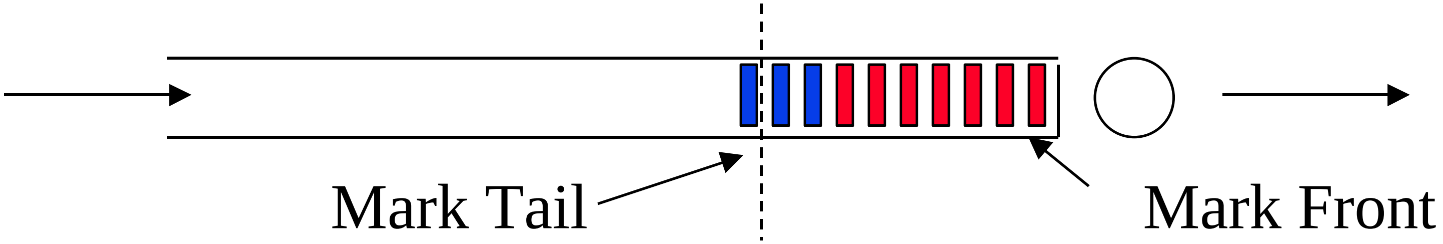
- ❑ Mark-Front improves the link efficiency for the same threshold.

Unfairness



- Mark-Front improves fairness

Task 4: Summary of Results

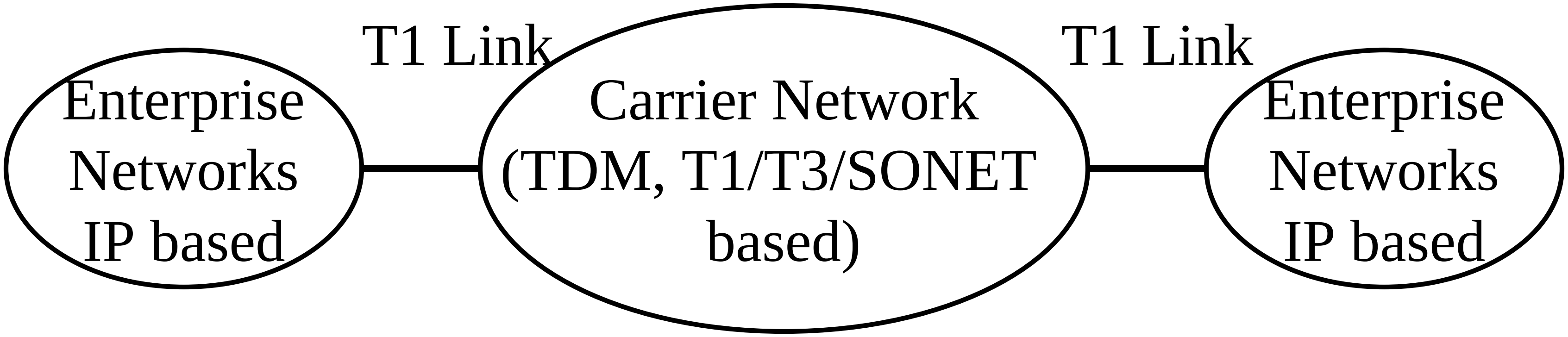


- Mark-front strategy
  - Reduces the buffer size requirement,
  - Increases link efficiency,
  - Avoids lock-in phenomenon and improves fairness
- This is specially important for long-bandwidth delay product networks

Task 5: IP Circuit Emulation

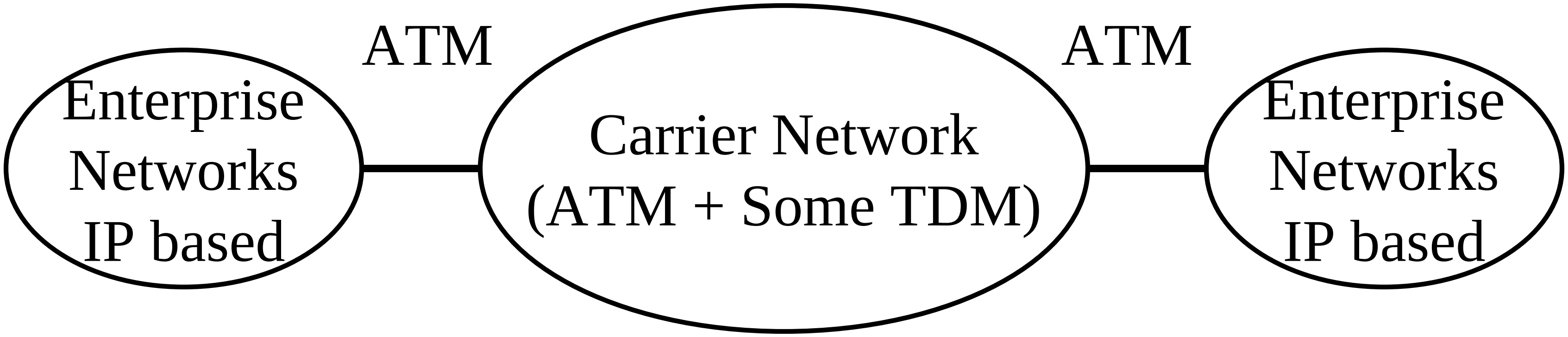
- Problem Statement
- Applicability

Telecommunications Today



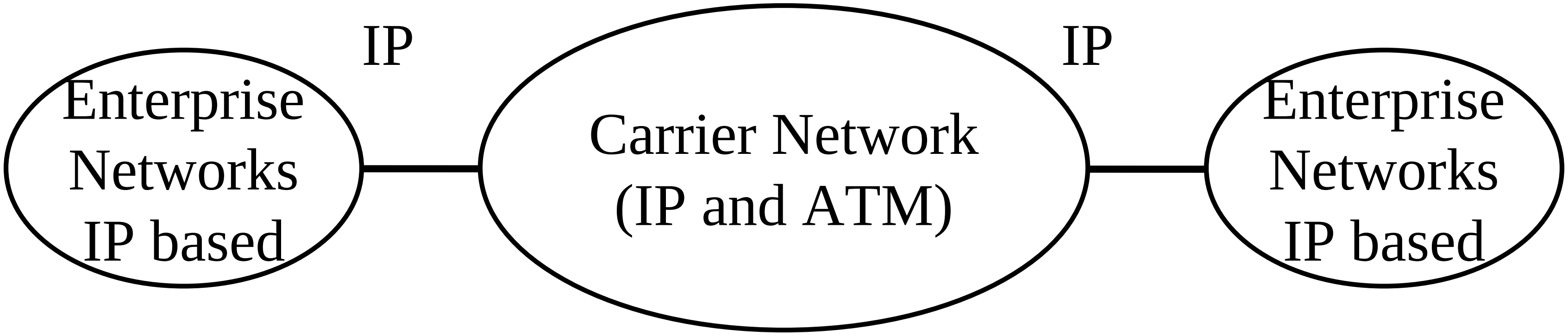
- Leased lines are a major source of revenue for carriers
- Both voice and data customers use leased lines

Telecommunications Tomorrow


ATM
ATM
Enterprise Networks IP based
Carrier Network (ATM + Some TDM)
Enterprise Networks IP based

- Carriers are switching to ATM
- ATM is not being adopted in the enterprise.
  ⇒ There is a pressure on carriers to provide native IP services

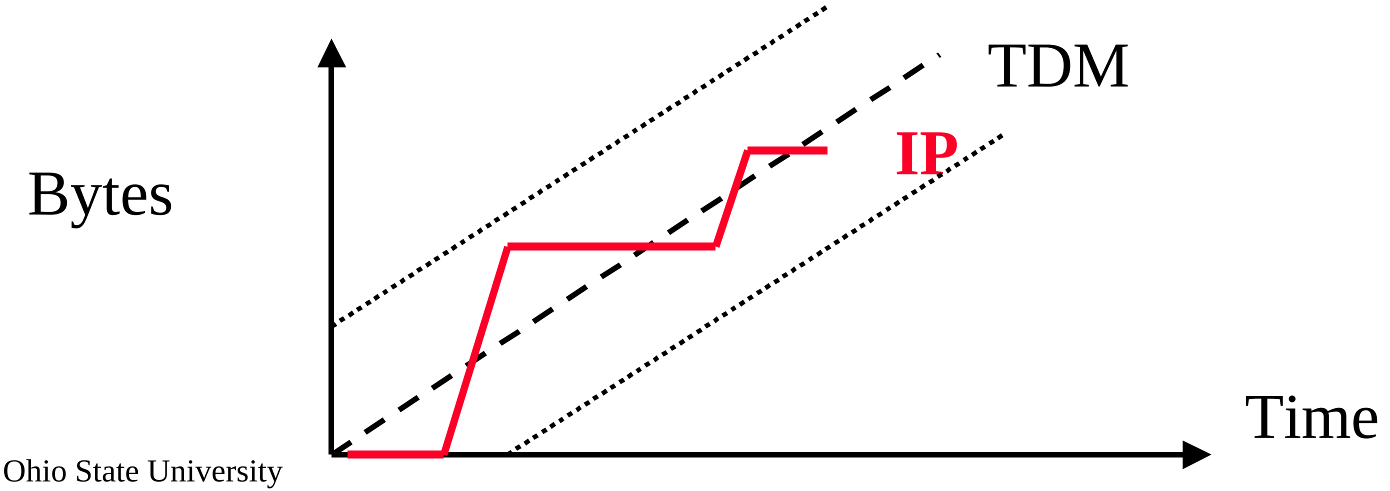
Telecommunications Future


IP
IP
Enterprise Networks IP based
Carrier Network (IP and ATM)
Enterprise Networks IP based

- ❑ Carriers will switch to IP routers and switches
- ❑ MPLS will provide the glue to connect ATM and IP equipment. (MPLS works with both technologies)
- ❑ It is important to provide leased line services to customers on both ATM and IP based equipment

CBR over IP

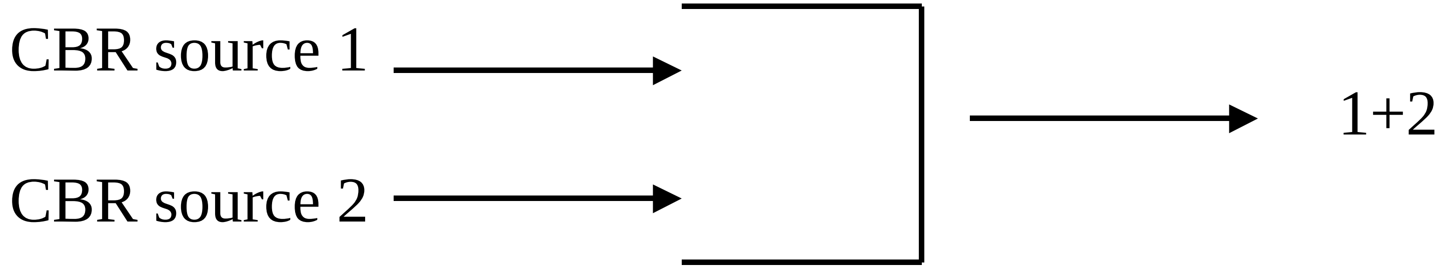
- Definition of CBR is not trivial
- In the TDM world: CBR = Fixed bits/sec
- In ATM world: Cells are fixed size but can arrive slightly at different time
  ⇒ CBR = Small CDVT (Cell delay variation Tolerance)
- In IP: Packets are of different size and bursty



CBR over IP (cont)

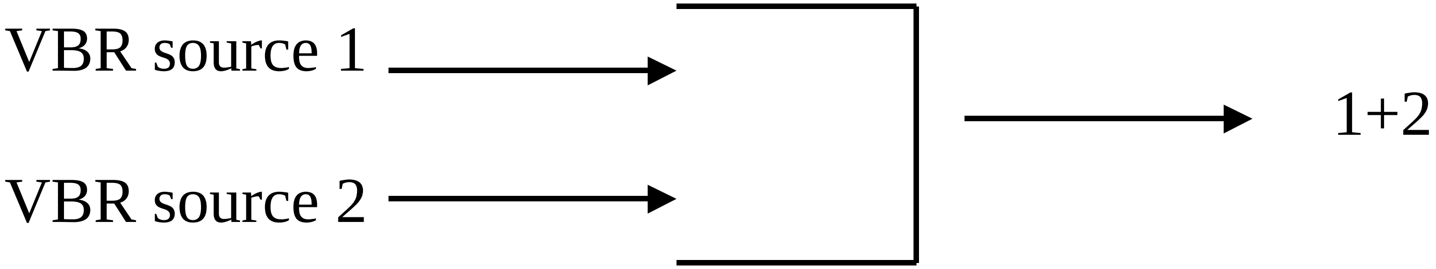
- CBR is defined by limits on burstiness and rate
- How much variation from the straight line is ok?
- What these limits are and how to best specify them is the first problem in this project. The solution has many requirements as indicated next (including aggregation for multiple sources).
- Traditional methods such as leaky bucket or token bucket are not additive and not good for highly bursty data traffic (ok for slightly bursty voice traffic)

Aggregation



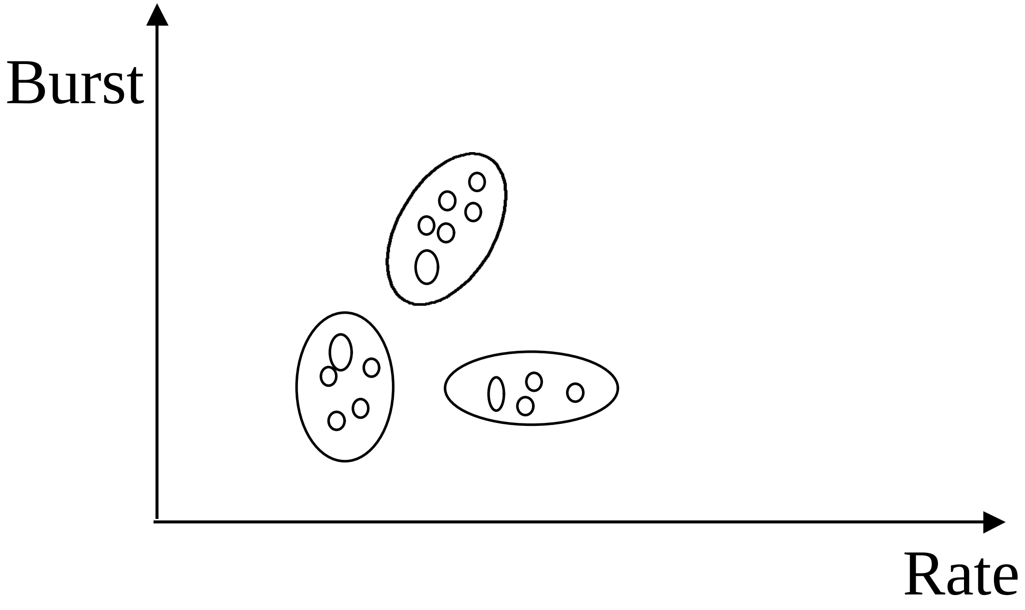
- How to combine multiple CBR sources?

Aggregation of VBR Sources



- How to combine multiple VBR sources?

Optimized Solution

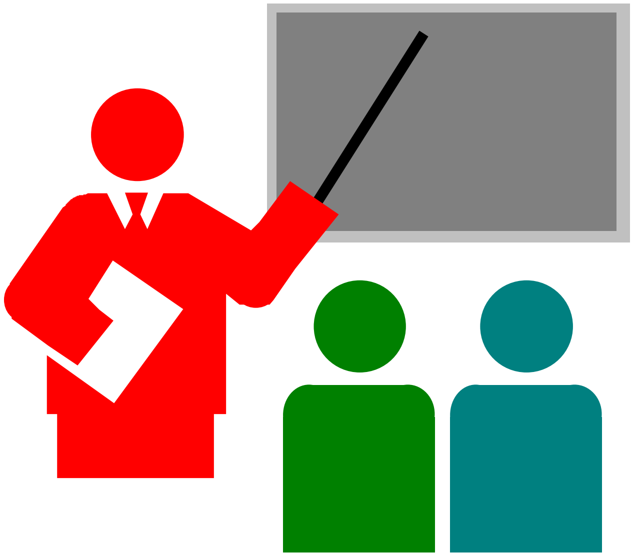


- Find the best rules how to multiplex different IP traffics

Current Work Status

- Using Leaky Bucket Model to do simulation
- Find the optimized multiplexing solution for different input traffics scenarios
  - For example: Multiplexing voice over IP
- Will work on D-BIND/H-BIND model later on
- Study to propose better model

Summary



1. Explicit congestion notification can be improved with mark front strategy
2. Circuit emulation requires better models for traffic specification

Thank You!

Final Presentation
at
NASA Glenn Research Center,
Cleveland, Ohio.

December 8, 2000

QoS for Real Time Applications over Next Generation Data Networks

Final Project Presentation
December 8, 2000
http://www.engr.udayton.edu/faculty/matiquzz/Pres/QoS-final.pdf

University of Dayton

Mohammed Atiquzzaman (PI)
Haowei Bai
Hongjun Su
Jyotsna Chitri
Faruque Ahamed

NASA Glenn

Will Ivancic

Progress of the tasks

- Progress to date:
  - Task 1: QoS in Integrated Services over DiffServ networks (UD)
  - Task 2: Interconnecting ATN with the next generation Internet (UD)
  - Task 3: QoS in DiffServ over ATM (UD)
  - Task 4: Improving Explicit Congestion Notification with the Mark-Front Strategy (OSU)
  - Task 5: Multiplexing VBR over VBR (OSU)
  - Task 6: Achieving QoS for TCP traffic in Satellite Networks with Differentiated Services (OSU)
- Conclusions

Task 1

QoS in *Integrated Services* over *DiffServ networks*

Integrated Services

- IntServ is one of models proposed by IETF to meet the demand for end-to-end QoS over heterogeneous networks.
- The IntServ model is characterized by resource reservation.
- IntServ implementation requires RSVP (Resource Reservation Protocol) signaling and resource allocations at every network element along the path.
- All the network elements must be RSVP-enable.
- Scalability problem becomes a bound on its implementation for the entire Internet.

Differentiated Services

- DiffServ is currently being standardized to overcome the scalability issue of IntServ.
- This model does not require significant changes to the existing infrastructure, and does not need many additional protocols.
- Because it is based on the processing of aggregated flows (classes), DiffServ does not consider the need of end applications.

IntServ vs. DiffServ

- Advantage of IntServ: application oriented.
- Disadvantage of IntServ: scalability problem.

- Advantage of DiffServ: enables scalability across large networks.
- Disadvantage of DiffServ: does not consider the QoS requirements of end users.

Problem Statement

- With the implementation of IntServ for small WAN networks and DiffServ for the Internet backbone, combining the advantages of IntServ and DiffServ, can TCP/IP traffic meet the QoS demands?
- Can QoS be provided to end applications?

Objective

- To investigate the POSSIBILITY of providing end-to-end QoS when IntServ runs over DiffServ backbone in the next generation Internet.
- To propose a MAPPING FUNCTION to run IntServ over the DiffServ backbone.
- To show the SIMULATION RESULTS used to prove that QoS can be achieved by end IntServ applications when running over DiffServ backbone in the next generation Internet.

Possibility of End-to-end QoS

- To support the end-to-end QoS model, the IntServ architecture must be supported over a wide variety of different types of network elements.
- In this context, a network that supports DiffServ may be viewed as a network backbone.
- Combining IntServ and DiffServ has been proposed by IETF as one of the possible solutions to achieve end-to-end QoS in the next generation Internet.

Possibility of End-to-end QoS (Continued)

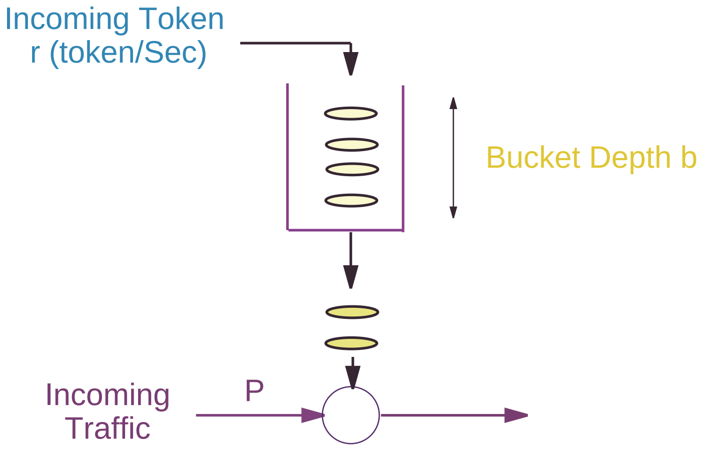
- Obviously, a mapping function from IntServ flows to DiffServ classes has to be developed.

Requirements of IntServ Services

- The objective of *Guaranteed Service*: to achieve a bounded delay, which means it does not control the minimal or average delay of datagram, merely the maximal delay.
- The objective of *Controlled-load Service:* to achieve little or no delay as that provided to best-effort traffic under lightly loaded conditions.
- Delay: fixed delay (such as transmission delay) & queuing delay.
- Fixed delay is a property of chosen path; queuing delay is primarily a function of token bucket and data rate.

Tspec

- ***Tspec***: according to RFC 2212 (Guaranteed service) and RFC 2211 (Controlled-load service), ***Tspec*** takes the form of:
  - a token bucket specification, i.e., bucket rate (*r*) and bucket depth (*b*), and
  - a peak rate (*p*), a minimum policed unit (*m*) and a maximum datagram size (*M*).



Resource Reservation

- In IntServ, resource reservation are made by requesting a service type specified by a set of quantitative parameters known as ***Tspec (Traffic Specification)***.

- Each set of parameters determines an appropriate priority level.

Mapping Considerations

- PHBs in DiffServ domain must be appropriately selected for each requested service in IntServ domain.
- The required policing, shaping and marking must be done at the edge router of the DiffServ domain.
- Taking into account the resource availability in DiffServ domain, admission control must be implemented for requested traffic in IntServ domain.

Proposed IntServ to DiffServ Mapping

- In this study, we propose to map
  - **Guaranteed service** to **EF** PHB and
  - **Controlled-load** service to **AF** PHBs.

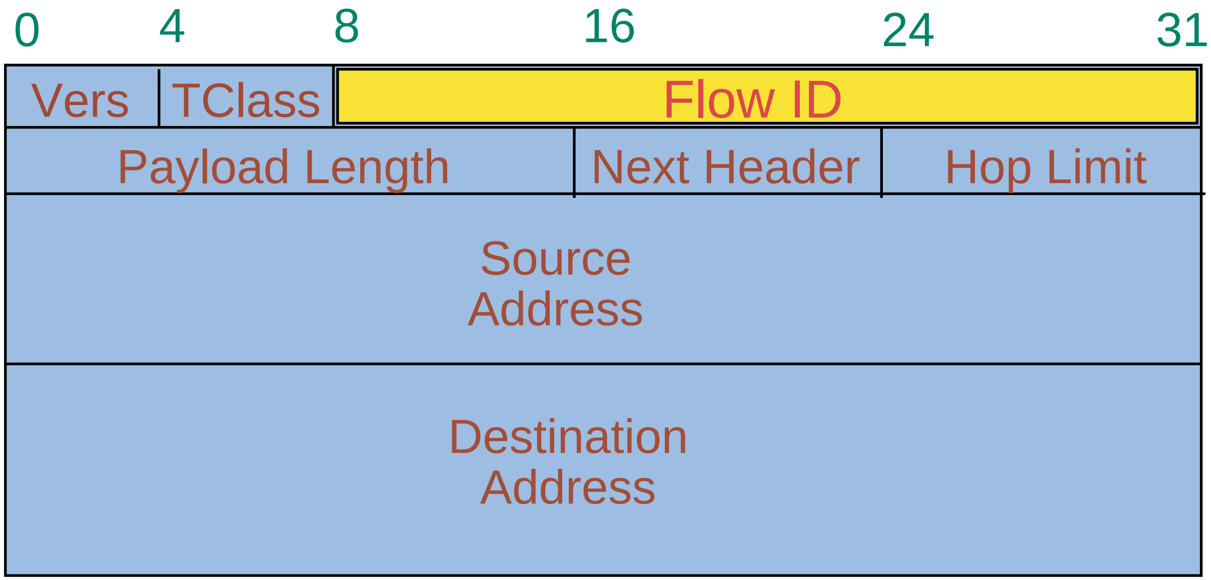
Mapping Function

- **Mapping function:** a function which is used to assign an appropriate DSCP (DiffServ Codepoint) to a flow specified by *Tspec* parameters in IntServ domain.

- The Mapping function should ensure that the required QoS could be achieved for IntServ when running over DiffServ domain.

Mapping Function (Continued)

- Each packet in the flow from the IntServ domain has a **flow ID** indicated by the value of *flow-id* field in the IP header.



0 4 8 16 24 31

Vers | TClass | Flow ID

Payload Length | Next Header | Hop Limit

Source Address

Destination Address

40-octet IPv6 base header

Mapping Function (Continued)

- The **flow ID** attributed with the *Tspec* parameters is used to determine which flow the packet belongs to.
- It is possible for different senders to use the same *Tspec* parameters to request service. However, they are differentiated by the **flow ID**. **Flow ID is unique.**
- It is also possible that different flows can be mapped to the same PHB in the DiffServ domain.

An Example Mapping Function

- An example mapping function used in our simulation

Tspec	***Flow ID***	***PHB***	***DSCP***
r=0.7 Mb, *b*=5000 bytes	0	EF	101110
r=0.7 Mb, *b*=5000 bytes	1	EF	101110
r=0.5 Mb, *b*=8000 bytes	2	AF11	001010
r=0.5 Mb, *b*=8000 bytes	3	AF11	001010
r=0.5 Mb, *b*=8000 bytes	4	AF11	001010

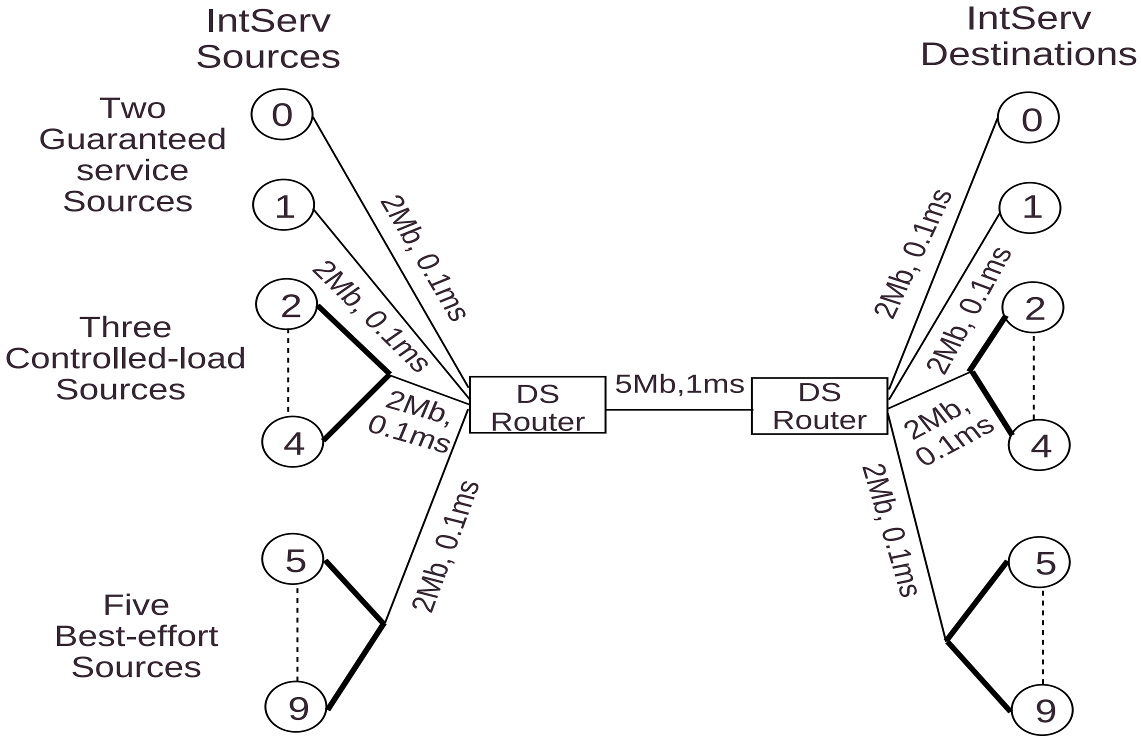
How Does the Mapping Function Work?

- Packets specified by *Tspec* parameters and *Flow ID* are first mapped to the corresponding PHBs in the DiffServ domain by appropriately assigning a DSCP according to the mapping function.

- Packets are then routed in the DiffServ domain where they receive treatments based on their DSCP code.

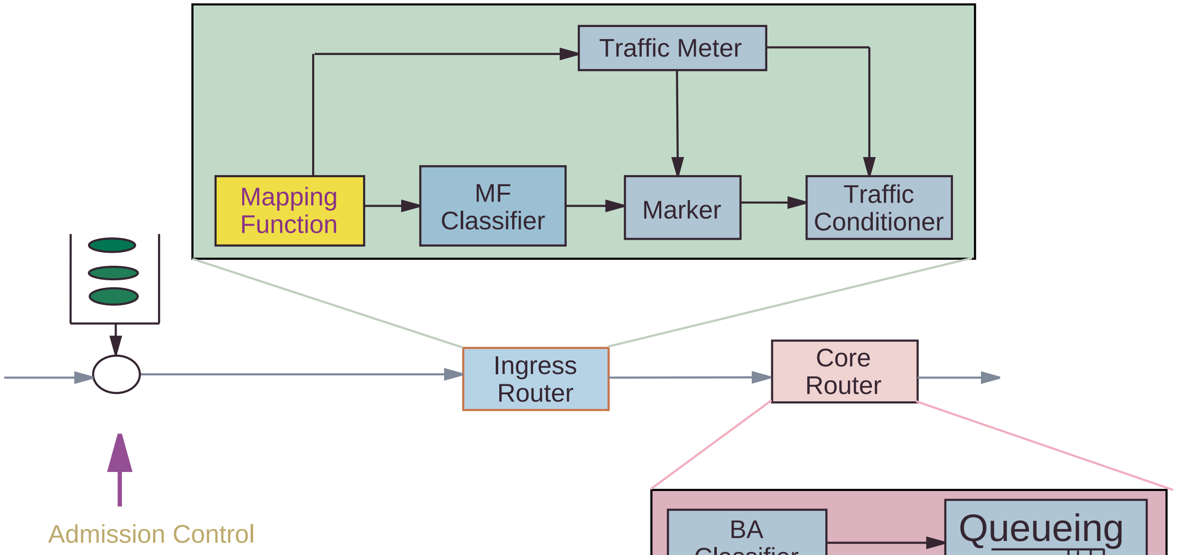
Simulation Configurations

- Simulation tool: Berkeley *ns* V2.1b6
- Simulation configuration.



Simulation Configuration (Continued)

- We integrated the mapping function into the edge DiffServ router.



Simulation Configurations (Continued)

- Configuration of queues inside the core DiffServ router.

	Queue Type	**Scheduler weight**
EF Queue	*PQ-Tail drop*	*0.4*
AF Queue	*RIO*	*0.4*
BE Queue	*RED*	*0.2*

- Since the bandwidth of bottleneck link is 5Mb, the above scheduling weight implies bandwidth of
  - EF: 2Mb
  - AF: 2Mb
  - BE: 1Mb

Performance Criteria

- **Goodput of each IntServ source**.
- **Queue size of each queue in the DiffServ core router**.
- **Drop ratio at scheduler**.
- **Non-conformant ratio**: the ratio of non-conformant packets as compared to in-profile packets.

QoS Obtained by Guaranteed Service: Case 1

- ***Case 1:*** **No congestion; no excessive traffic.**

Source NO.	**Source Type**	**Source Rate**	**Resource Reservation**	
			Token Rate	**Bucket Depth**
0	*Guaranteed Service*	*0.7Mb*	*0.7Mb*	*5000bytes*
1	*Guaranteed Service*	*0.7Mb*	*0.7Mb*	*5000bytes*

- Two Guaranteed service sources generate 1.4Mb traffic which is less than the scheduled bandwidth (2Mb) → **No congestion**.
- Source rate is equal to the bucket rate → **No excessive traffic**.

Goodput of Guaranteed Service: Case 1

- ***Simulation results of Case 1:*** **Goodput of each Guaranteed Service source.**

Source No.	**Flow ID**	**Case 1 (Kb/S)**
0	*0*	*699.82*
1	*1*	*699.80*

- **Observation:** the goodput of each source is almost equal to the corresponding source rate.

Drop Ratio of Guaranteed Service: Case 1

- ***Simulation results of Case 1:*** **Drop ratio of Guaranteed Service traffic (measured at scheduler).**

Type of traffic	Case 1
Guaranteed Service	*0.00*

- **Observation:** since there is no significant congestion, the drop ratio is zero.

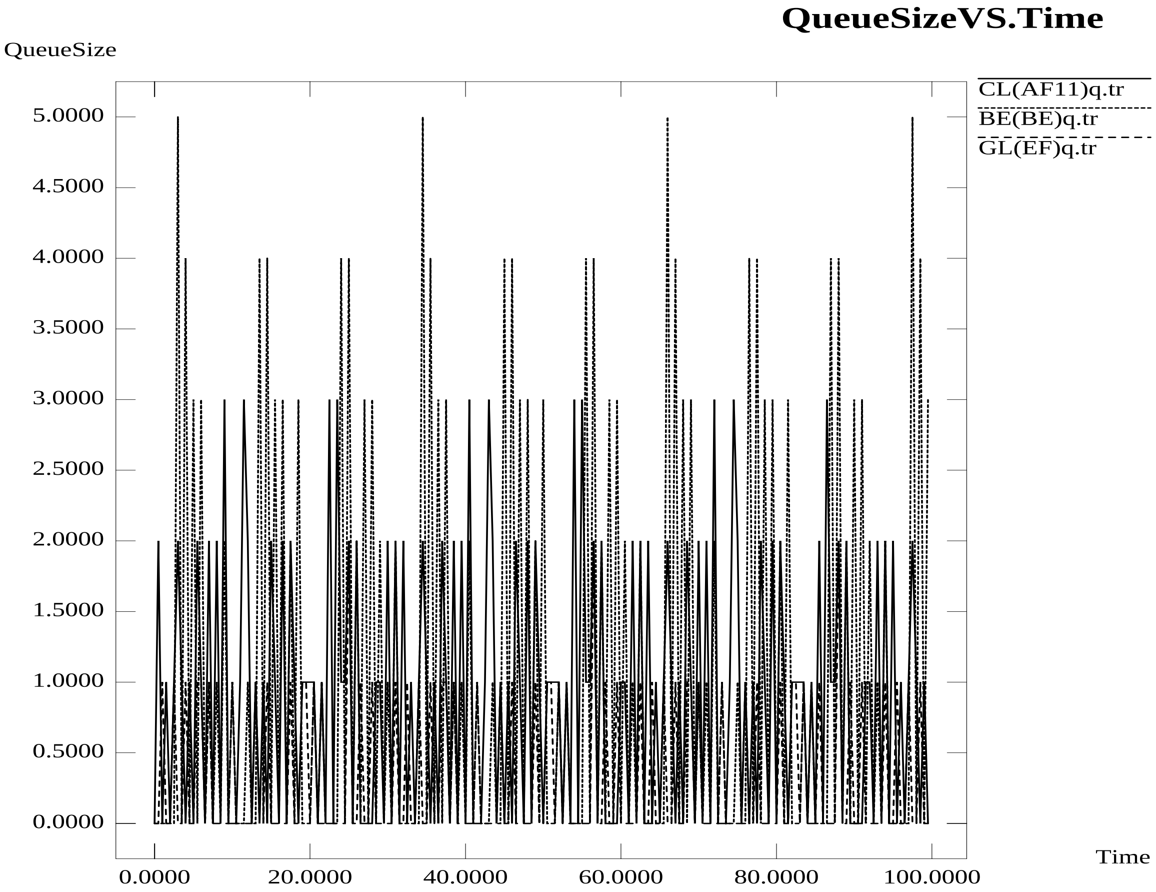
Non-conformant Ratio of Guaranteed Service: Case 1

- ***Simulation results of Case 1:*** **non-conformant ratio of each Guaranteed Service source.**

Source No.	Flow ID	Case 1
0	*0*	*0.00*
1	*1*	*0.00*

- **Observations:** since there is no excessive traffic, the non-conformant ratio is zero.

Queue Size Plot: Case 1



Queue Size Plot: Case 1 (Continued)

- **Observations:**
  - Since Case 1 is an ideal case, the size of each queue is very small.
  - BE queue has the largest jitter.

QoS Obtained by Guaranteed Service: Case 2

- ***Case 2:*** **No congestion; source 1 generates excessive traffic.**

Source NO.	Source Type	Source Rate	Resource Reservation	
			Token Rate	Bucket Depth
0	*Guaranteed Service*	*0.7Mb*	*0.7Mb*	*5000bytes*
1	*Guaranteed Service*	*0.9Mb*	*0.7Mb*	*5000bytes*

- Two Guaranteed service sources generate 1.6Mb traffic which is less than the scheduled bandwidth (2Mb) → **No congestion**.
- Source rate of source 1 (0.9 Mb) is greater than the bucket rate (0.7Mb) → **Source 1 generates excessive traffic**.

Goodput of Guaranteed Service: Case 2

- ***Simulation results of Case 2:*** **Goodput of each Guaranteed Service source.**

Source No.	Flow ID	Case 1 (Kb/S)	Case 2 (Kb/S)
0	*0*	*699.83*	*699.80*
1	*1*	*699.80*	*699.64*

- **Observations:** the goodput of source 0 is almost equal to the corresponding source rate; *however, the goodput of source 1 is equal to its token rate, 0.7Mb, instead of its source rate, 0.9Mb.*

Drop Ratio of Guaranteed Service: Case 2

- ***Simulation results of Case 2:*** **Drop ratio of Guaranteed Service traffic (measured at scheduler).**

Type of traffic	**Case 1**	**Case 2**
Guaranteed Service	*0.00*	*0.00*

- **Observation:** since there is no significant congestion, the drop ratio is zero.

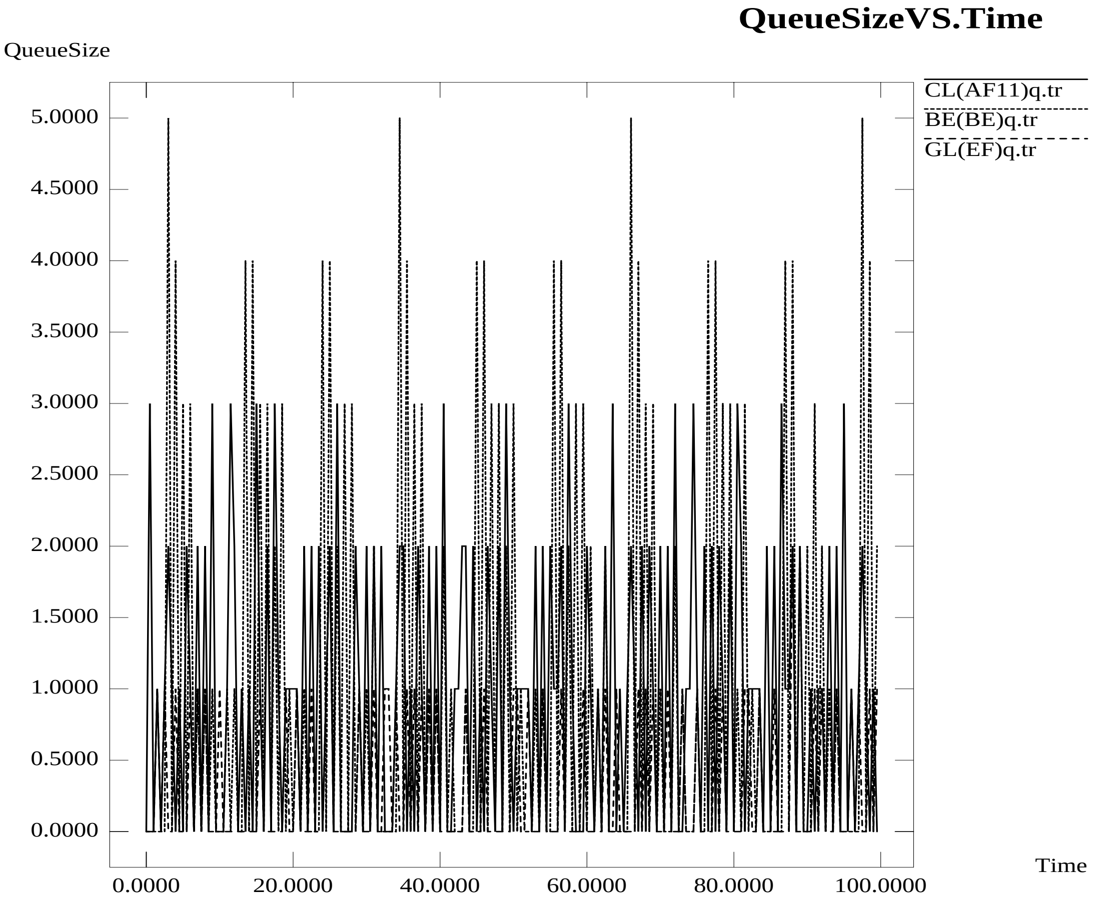
Non-conformant Ratio Guaranteed Service: Case 2

- ***Simulation results of Case 2:*** **non-conformant ratio of each Guaranteed Service source.**

Source No.	**Flow ID**	**Case 1**	**Case 2**
0	*0*	*0.00*	*0.00*
1	*1*	*0.00*	*0.22*

- **Observation:** since source 1 generates excessive traffic, its non-conformant ratio is increased, compared to Case 1.

Queue Size Plot: Case 2



Queue Size Plot: Case 2 (Continued)

- **Observations:**
  - In this case, BE queue has the largest size and jitter;
  - Guaranteed service queue has the smallest size and jitter.
  - In addition, compared with Case 1, the upper bound of Guaranteed queue size is guaranteed.

QoS Obtained by Guaranteed Service: Case 3

- ***Case 3:*** **Guaranteed service gets into congestion; no excessive traffic----evaluation under congestion.**

Source NO.	**Source Type**	**Source Rate**	**Resource Reservation**	
			Token Rate	**Bucket Depth**
0	*Guaranteed Service*	*0.7Mb*	*0.7Mb*	*5000bytes*
1	*Guaranteed Service*	*2Mb*	*2Mb*	*5000bytes*

- Two Guaranteed service sources generate 2.7Mb traffic which is greater than the scheduled bandwidth (2Mb) → **Guaranteed service gets into congestion**.
- Source rate is equal to the bucket rate → **No excessive traffic**.

Goodput of Guaranteed Service: Case 3

- *Simulation results of Case 3:* **Goodput of each Guaranteed Service source.**

Source No.	Flow ID	Case 1 (Kb/S)	Case 2 (Kb/S)	Case 3 (Kb/S)
0	*0*	*699.8250*	*699.8039*	*459.8790*
1	*1*	*699.8039*	*699.6359*	*1540.1400*

- **Observation:** the total goodput of two sources is limited by the scheduled bandwidth, 2Mb, instead of 2.7Mb.

Drop Ratio of Guaranteed Service: Case 3

- ***Simulation results of Case 3:*** **Drop ratio of Guaranteed Service traffic (measured at scheduler).**

Type of traffic	Case 1	Case 2	Case 3
Guaranteed Service	*0.00*	*0.00*	*0.26*

- **Observation:** the drop ratio is increased.

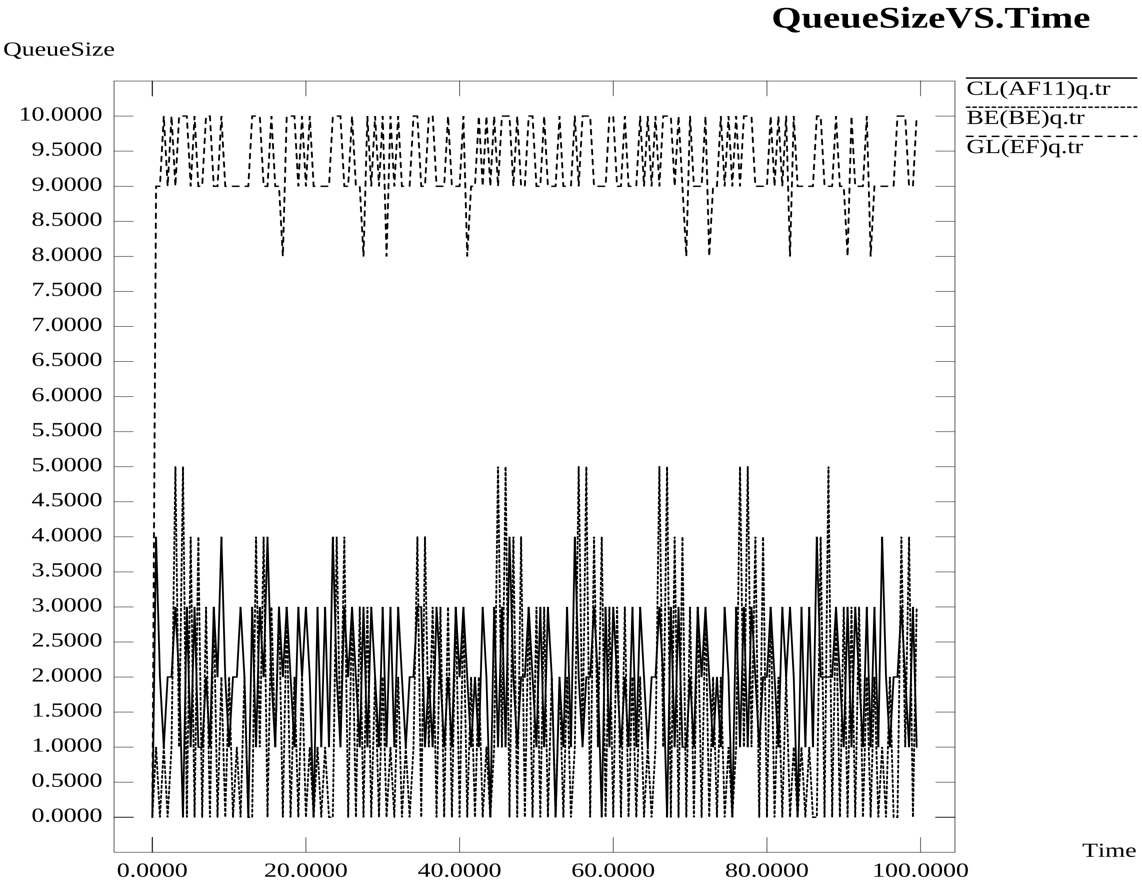
Non-conformant Ratio of Guaranteed Service: Case 3

- ***Simulation results of Case 3:*** **non-conformant ratio of each Guaranteed Service source.**

Source No.	Flow ID	Case 1	Case 2	Case 3
0	*0*	*0.00*	*0.00*	*0.00*
1	*1*	*0.00*	*0.22*	*0.00*

- **Observations:** since there is no excessive traffic, the non-conformant ratio is zero.

Queue Size Plot: Case 3



Queue Size Plot: Case 3 (Continued)

- **Observations:** since we increased the source rate and token rate of source 1 in order to make Guaranteed service congested, it is reasonable that the upper bound of Guaranteed service queue size is increased.

QoS Obtained by Controlled-load Service

- **No congestion; source 3 generates excessive traffic. (Similar to Case 2 of Guaranteed Service)**

Source NO.	**Source Type**	**Source Rate**	**Resource Reservation**	
			Token Rate	**Bucket Depth**
2	*Controlled-load Service*	*0.5Mb*	*0.5Mb*	*8000bytes*
3	*Controlled-load Service*	*0.7Mb*	*0.5Mb*	*8000bytes*
4	*Controlled-load Service*	*0.5Mb*	*0.5Mb*	*8000bytes*

- Total source rate is 1.7Mb, less than the scheduled bandwidth (2Mb) → **No congestion**.
- Source rate of source 3 (0.7 Mb) is greater than its bucket rate (0.5Mb) → **Source 3 generates excessive traffic**.

Goodput of Controlled-load Service

- ***Simulation results:*** **Goodput of each Guaranteed Service source.**

Source No.	Flow ID	Goodput (Kb/S)
2	*2*	*499.9889*
3	*3*	*700.0140*
4	*4*	*499.9889*

- **Observations:** the goodput of each source is almost equal to the corresponding source rate, *which is different from Case 2 of Guaranteed service.*
- This is because the non-conformant packets are degraded and then forwarded. (Proposed as one of the forwarding scheme for non-conformant packets by RFC2211)

Drop Ratio of Controlled-load Service

- ***Simulation results:*** **Drop ratio of Guaranteed Service traffic (measured at scheduler).**

Type of traffic	**Drop ratio**
Controlled-load Service	*0.0000*

- **Observations:** since there is no significant congestion, the drop ratio is zero.

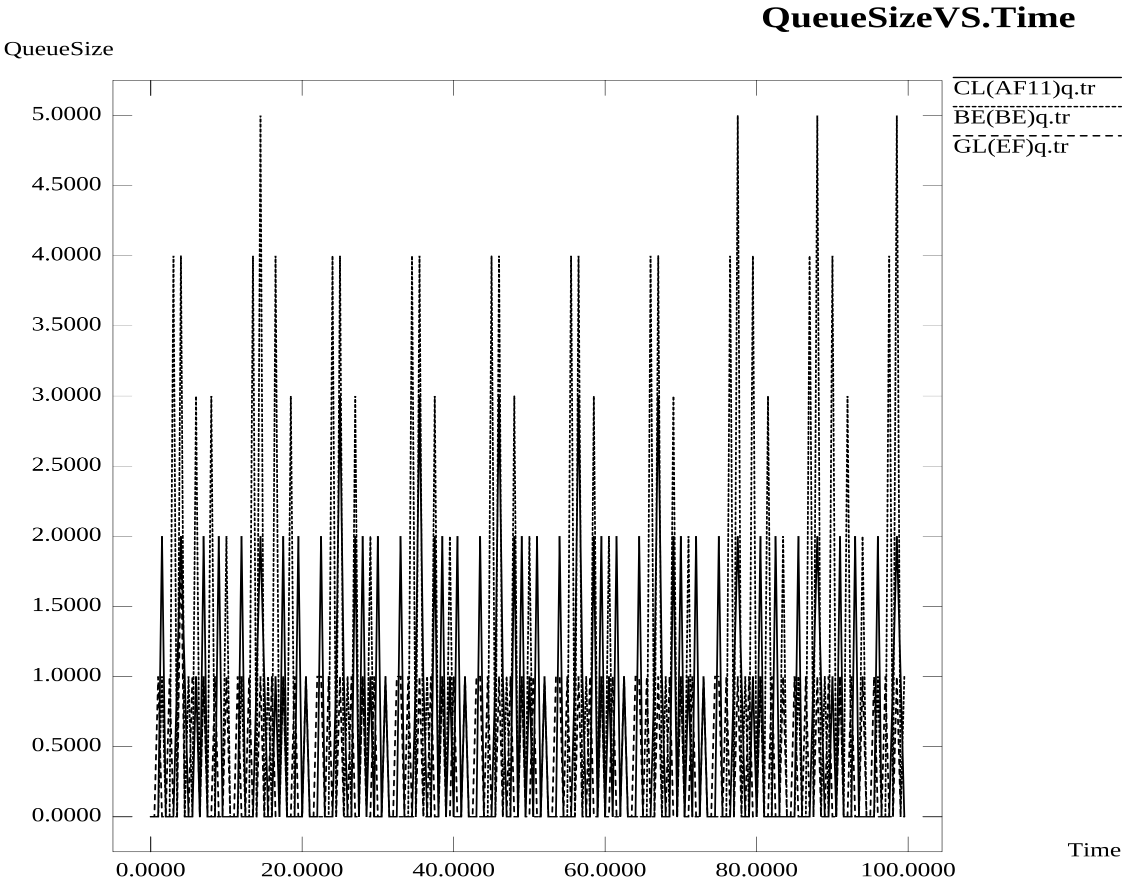
Non-conformant Ratio of Controlled-load Service

- ***Simulation results:*** **non-conformant ratio of each Guaranteed Service source.**

Source No.	**Flow ID**	**Non-conformant Ratio**
2	*2*	*0.00000*
3	*3*	*0.28593*
4	*4*	*0.00000*

- **Observations:** since source 3 generates excessive traffic, its non-conformant ratio is much higher than other two's.

Queue Size Plot



Observations

- The upper bound of queuing delay of ***Guaranteed Service*** is guaranteed. In addition, it always has the smallest jitter without being affected by other traffic flows.

- The ***controlled-load Service*** has the smaller jitter and queue size than the best effort traffic. The non-conformant packets are degraded and then forwarded.

Conclusion

- **We have shown that the end-to-end QoS requirements of IntServ applications can be successfully achieved when IntServ traffic is mapped to the DiffServ domain in the next generation Internet.**

Task 2

Interconnecting *ATN* with *Next Generation Internet*

QoS Requirements of ATN

- To carry time critical information required for aeronautical applications, ATN provides different QoS to applications.
- In the ATN, priority has the essential role of ensuring that high priority safety related and time critical data are not delayed by low priority non-safety data, especially when the network is overloaded with low priority data.
- The time critical information carried by ATN and the QoS required by ATN applications has led to the development of the ATN as an expensive independent network.

ATN & DiffServ

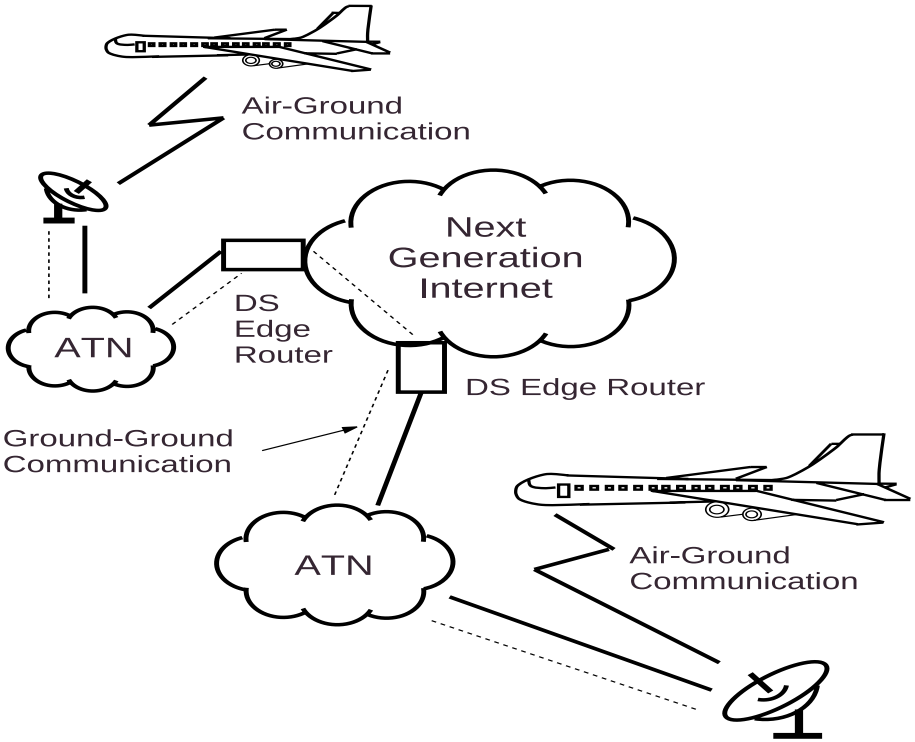
- The largest public network, Internet, only offers best-effort service to users and hence is not suitable for carrying time critical ATN traffic.
- The rapid commercialization of the Internet has given rise to demands for QoS over the Internet.
- DiffServ has been proposed by IETF as one of models to meet the demand for QoS.

Objective

- To investigate the POSSIBILITY of providing QoS to ATN applications when it runs over DiffServ backbone in the next generation Internet.
- To propose a MAPPING FUNCTION to run ATN over the DiffServ backbone.
- To show the SIMULATION RESULTS used to prove that QoS can be achieved by end ATN applications when running over DiffServ backbone in the next generation Internet.

Significance

- Considerable cost savings could be possible if the next generation Internet backbone can be used to connect ATN subnetworks.

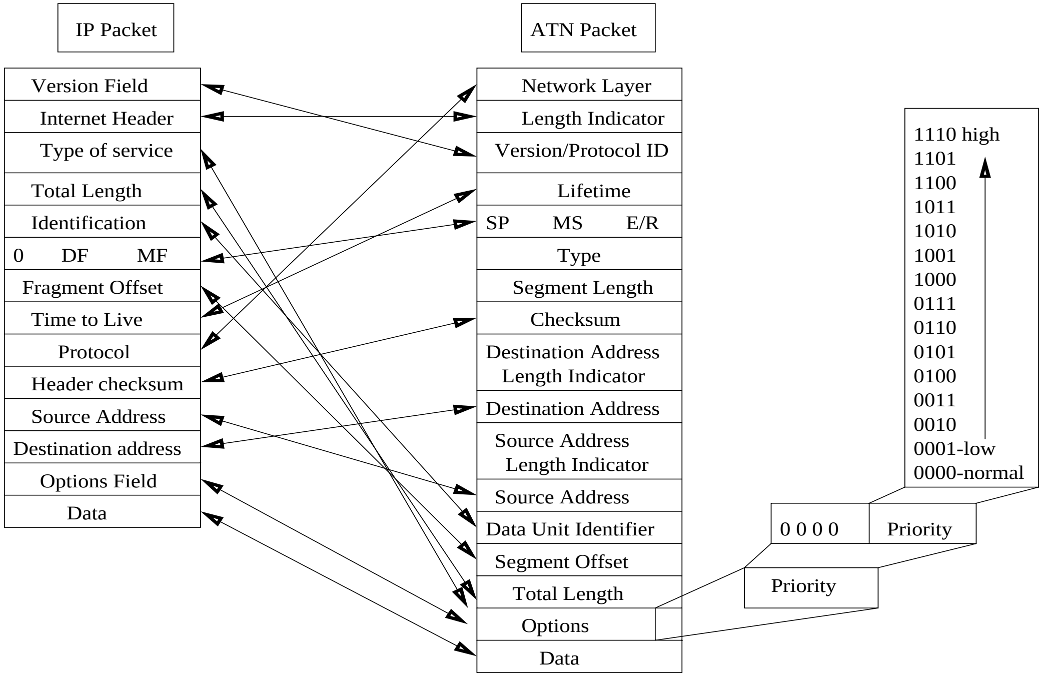


Possibility of ATN over DiffServ

- The DiffServ model utilizes **six bits in the TOS** (Type of Service) field of the IP header to mark a packet for being eligible for a particular forwarding behavior.
- The NPDU (Network Protocol Data Unit) header of an ATN packet contains an option part including **an 8-bit field named *Priority*** which indicates the relative priority of the NPDU.
- The value ***0000 0000*** indicates ***normal priority***; the values ***0000 0001* through *0000 1110*** indicate *the **priority in an increasing order.***

Possibility of ATN over DiffServ (Continued)

- The similarity between an ATN packet and an IP packet, shown below, provides the **possibility** for mapping ATN to DiffServ to achieve the required QoS when they are interconnected.



Mapping Consideration

- The PHB treatment of packets along the path in the DiffServ domain must approximate the QoS offered in the ATN network.

Mapping Function

- We map the ***normal priority*** (indicated by *Priority* field in NPDU) in ATN domain to ***BE*** PHB in DiffServ domain;
- Map the ***high priority*** in ATN domain to ***EF*** PHB in DiffServ domain;
- Map the ***medium priorities*** in ATN domain to the corresponding classes of ***AF*** PHBs in DiffServ domain.

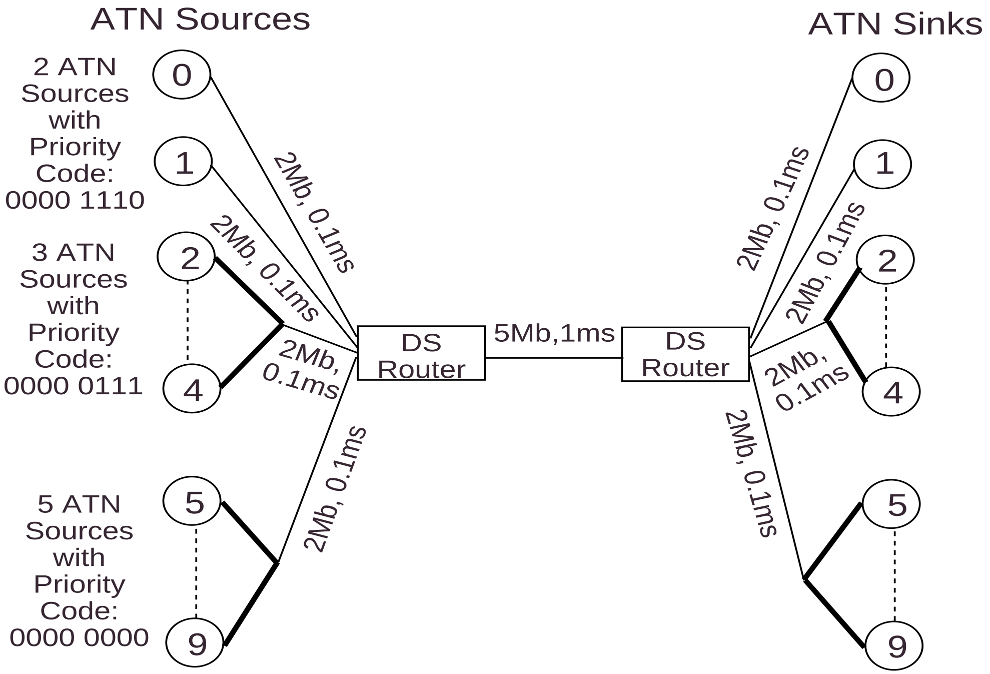
An Example Mapping Function

- An example mapping function used in our simulation

ATN Priority Code	***Priority***	***PHB***	***DSCP***
0000 0000	*Normal*	*BE*	*000000*
0000 0111	*Medium*	*AF11*	*001010*
00001110	*High*	*EF*	*101110*

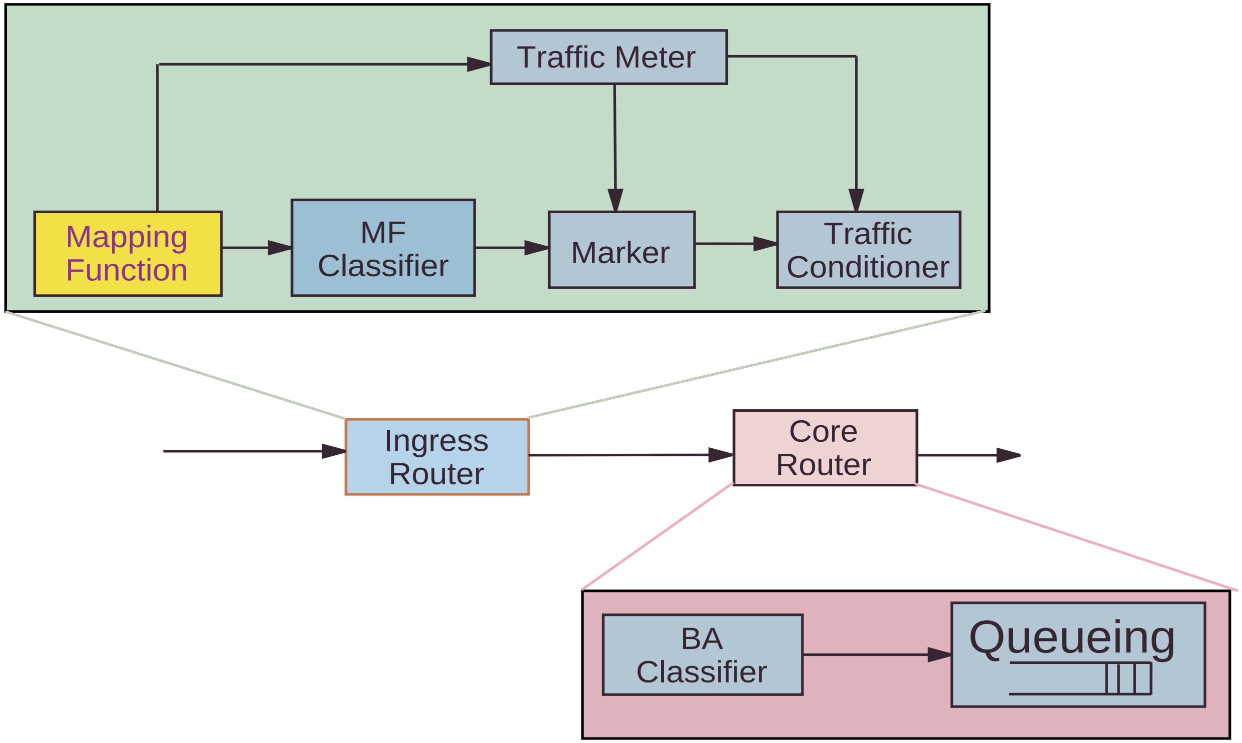
Simulation Configurations

- Simulation tool: Berkeley *ns* V2.1b6
- Simulation configuration.



Simulation Configurations (Continued)

- We integrated the mapping function into the edge DiffServ router. (***Recall***)



Simulation Configurations (Continued)

- Table below shows the configuration of queues inside the core DiffServ router.

	Queue Type	**Queue weight**
EF Queue	*PQ-Tail drop*	*0.4*
AF Queue	*RIO*	*0.4*
BE Queue	*RED*	*0.2*

Since the bandwidth of bottleneck link is 5Mb, the above scheduling weight implies bandwidth of

- EF: 2Mb
- AF: 2Mb
- BE: 1Mb

Performance Criteria

- Goodput of each ATN source.
- Queue size of each queue.
- Drop ratio at scheduler.

QoS Obtained by ATN Applications: Case 1

- ***Case 1:* No congestion.**

Source NO.	Source Type	Source Rate
0, 1	*High Priority*	*1Mb*
2, 3, 4	*Medium Priority*	*0.666Mb*
5,6,7,8,9	*Normal Priority*	*0.2Mb*

- The amount of traffic with different priorities are equal to the corresponding scheduled link bandwidth → **No congestion**.

Goodput of ATN Applications: Case 1

- ***Results of Case 1:*** **Goodput of each ATN source.**

Source Priority		Case 1 (Kb/S)	Case 2 (Kb/S)	Case 3 (Kb/S)	Case 4 (Kb/S)
High	*Src 0*	*999.99*	*999.99*	*999.99*	*999.99*
	Src 1	*999.99*	*999.99*	*999.99*	*999.99*
Medium	*Src 2*	*666.66*	*666.66*	*668.24*	*668.47*
	Src 3	*666.66*	*666.66*	*667.34*	*667.53*
	Src 4	*666.66*	*666.66*	*664.42*	*663.99*
Normal	*Src 5*	*200.00*	*199.65*	*200.00*	*199.48*
	Src 6	*200.00*	*201.85*	*200.00*	*201.98*
	Src 7	*200.00*	*202.42*	*200.00*	*201.68*
	Src 8	*199.98*	*199.88*	*199.98*	*200.467*
	Src 9	*200.00*	*196.20*	*200.00*	*196.39*

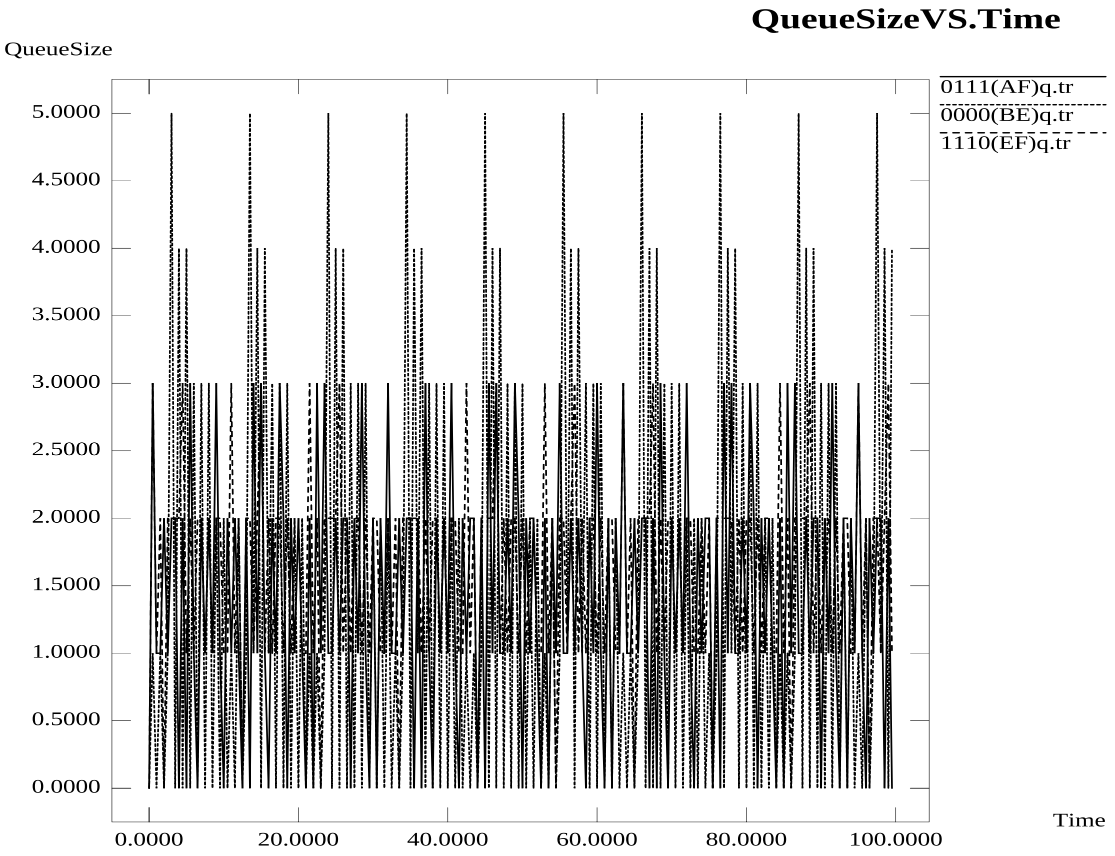
Drop Ratio of ATN Applications: Case 1

- ***Simulation results of Case 1:*** **Drop ratio of ATN traffic (measured at scheduler).**

Type of traffic	Case 1	Case 2	Case 3	Case 4
High Priority Traffic	*0.00*	*0.00*	*0.00*	*0.00*
Medium Priority Traffic	*0.00*	*0.00*	*0.49*	*0.49*
Normal Priority Traffic	*0.00*	*0.67*	*0.00*	*0.67*

- **Observations:** since there is no significant congestion, the drop ratio is zero.

Queue Size Plot: Case 1



Queue Size Plot: Case 1 (Continued)

- **Observations:** since Case 1 is a ideal case, the average size of each queue is very small. BE queue has the largest jitter.

QoS Obtained by ATN Applications: Case 2

- *Case 2: Normal Priority traffic gets into* **congestion.**

Source NO.	Source Type	Source Rate
0, 1	*High Priority*	*1Mb*
2, 3, 4	*Medium Priority*	*0.666Mb*
5,6,7,8,9	*Normal Priority*	*0.6Mb*

- The amount of traffic with Normal Priority (3Mb) is greater than the corresponding scheduled link bandwidth (1 Mb) → **Normal Priority traffic gets into congestion**.

Goodput of ATN Applications: Case 2

- ***Results of Case 2:*** **Goodput of each ATN source.**

Sources		Case 1 (Kb/S)	Case 2 (Kb/S)	Case 3 (Kb/S)	Case 4 (Kb/S)
High	*Src 0*	*999.9990*	*999.9990*	*999.9990*	*999.9990*
	Src 1	*999.9990*	*999.9990*	*999.9990*	*999.9990*
Medium	*Src 2*	*666.6660*	*666.6660*	*668.2409*	*668.4719*
	Src 3	*666.6660*	*666.6660*	*667.3379*	*667.5270*
	Src 4	*666.6660*	*666.6660*	*664.4189*	*663.9990*
Normal	*Src 5*	*200.0039*	*199.6469*	*200.0039*	*199.4790*
	Src 6	*200.0039*	*201.8520*	*200.0039*	*201.9780*
	Src 7	*200.0039*	*202.4190*	*200.0039*	*201.6840*
	Src 8	*199.9830*	*199.8779*	*199.9830*	*200.4660*
	Src 9	*200.0039*	*196.2030*	*200.0039*	*196.3920*

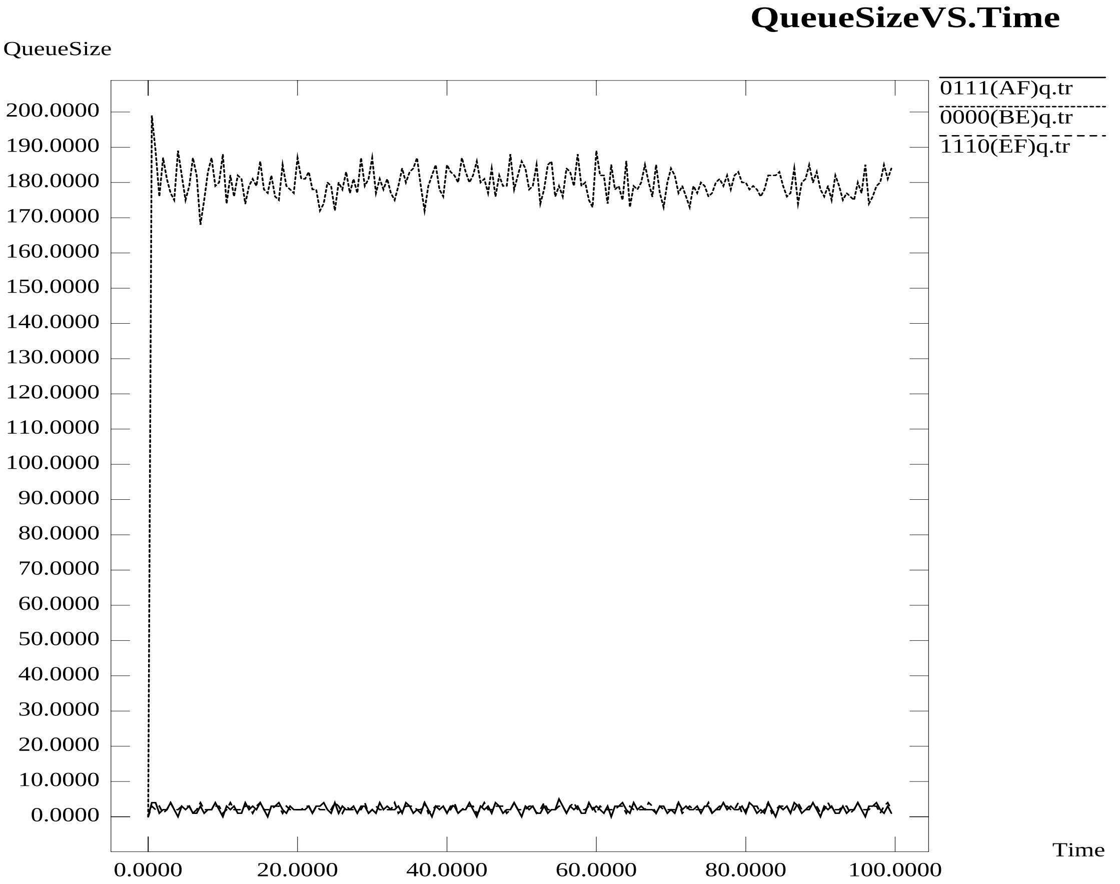
Drop Ratio of ATN Applications: Case 2

- ***Simulation results of Case 2:*** **Drop ratio of ATN traffic (measured at scheduler).**

Type of traffic	**Case 1**	**Case 2**	**Case 3**	**Case 4**
High Priority Traffic	*0.00000*	*0.00000*	*0.00000*	*0.00000*
Medium Priority Traffic	*0.00000*	*0.00000*	*0.49982*	*0.49982*
Normal Priority Traffic	*0.00000*	*0.66564*	*0.00000*	*0.66562*

- **Observations:** the drop ratio of Normal Priority traffic is increased.

Queue Size Plot: Case 2



Queue Size Plot: Case 2 (Continued)

- **Observations:**
  - In this case, the high priority traffic has the smallest average queue size and jitter;
  - The normal priority traffic has the biggest average queue size and jitter.

QoS Obtained by ATN Applications: Case 3

- ***Case 3: Medium Priority traffic gets into* congestion.**

Source NO.	**Source Type**	**Source Rate**
0, 1	*High Priority*	*1Mb*
2, 3, 4	*Medium Priority*	*1.333Mb*
5,6,7,8,9	*Normal Priority*	*0.2Mb*

- The amount of traffic with Medium Priority (4Mb) is greater than the corresponding scheduled link bandwidth (2 Mb) → **Medium Priority traffic gets into congestion**.

Goodput of ATN Applications: Case 3

- ***Results of Case 3:*** **Goodput of each ATN source.**

Sources		Case 1 (Kb/S)	Case 2 (Kb/S)	Case 3 (Kb/S)	Case 4 (Kb/S)
High	*Src 0*	*999.9990*	*999.9990*	*999.9990*	*999.9990*
	Src 1	*999.9990*	*999.9990*	*999.9990*	*999.9990*
Medium	*Src 2*	*666.6660*	*666.6660*	*668.2409*	*668.4719*
	Src 3	*666.6660*	*666.6660*	*667.3379*	*667.5270*
	Src 4	*666.6660*	*666.6660*	*664.4189*	*663.9990*
Normal	*Src 5*	*200.0039*	*199.6469*	*200.0039*	*199.4790*
	Src 6	*200.0039*	*201.8520*	*200.0039*	*201.9780*
	Src 7	*200.0039*	*202.4190*	*200.0039*	*201.6840*
	Src 8	*199.9830*	*199.8779*	*199.9830*	*200.4660*
	Src 9	*200.0039*	*196.2030*	*200.0039*	*196.3920*

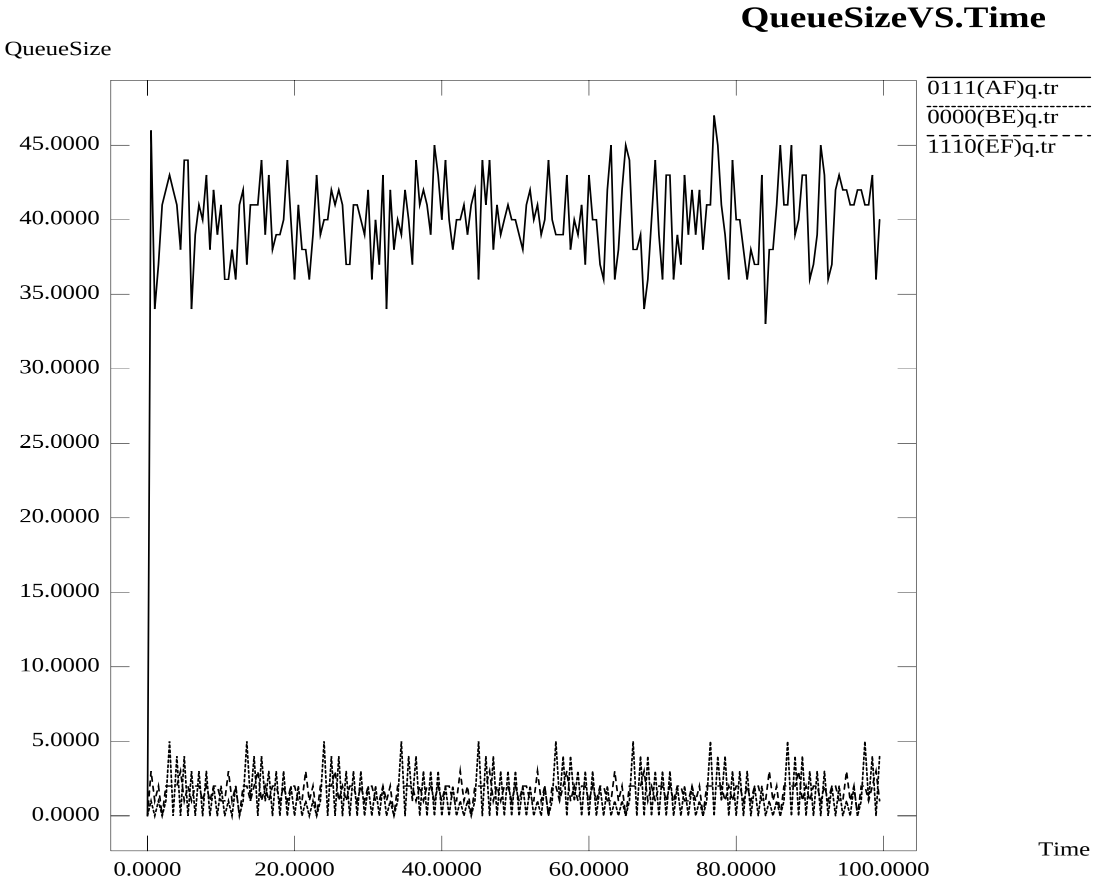
Drop Ratio of ATN Applications: Case 3

- ***Simulation results of Case 3:*** **Drop ratio of ATN traffic (measured at scheduler).**

Type of traffic	Case 1	Case 2	Case 3	Case 4
High Priority Traffic	*0.00000*	*0.00000*	*0.00000*	*0.00000*
Medium Priority Traffic	*0.00000*	*0.00000*	*0.49982*	*0.49982*
Normal Priority Traffic	*0.00000*	*0.66564*	*0.00000*	*0.66562*

- **Observations:** the drop ratio of Medium Priority traffic is increased.

Queue Size Plot: Case 3



Queue Size Plot: Case 3 (Continued)

- **Observations:**
  - The high priority traffic has the smallest average queue size and jitter.
  - Note that both the queue size and jitter of medium priority traffic are greater than the other two’s.

QoS Obtained by ATN Applications: Case 4

- ***Case 4: Both Medium and Normal Priority traffic gets into*** **congestion.**

Source NO.	Source Type	Source Rate
0, 1	*High Priority*	*1Mb*
2, 3, 4	*Medium Priority*	*1.333Mb*
5,6,7,8,9	*Normal Priority*	*0.6Mb*

- The amount of traffic with both Medium (4Mb) and Normal Priority (3Mb) is greater than the corresponding scheduled link bandwidth (2Mb, 1Mb) → **Both Medium and Normal Priority traffic gets into congestion**.

Goodput of ATN Applications: Case 4

- ***Results of Case 4:*** **Goodput of each ATN source.**

Sources		Case 1 (Kb/S)	Case 2 (Kb/S)	Case 3 (Kb/S)	Case 4 (Kb/S)
High	*Src 0*	*999.9990*	*999.9990*	*999.9990*	*999.9990*
	Src 1	*999.9990*	*999.9990*	*999.9990*	*999.9990*
Medium	*Src 2*	*666.6660*	*666.6660*	*668.2409*	*668.4719*
	Src 3	*666.6660*	*666.6660*	*667.3379*	*667.5270*
	Src 4	*666.6660*	*666.6660*	*664.4189*	*663.9990*
Normal	*Src 5*	*200.0039*	*199.6469*	*200.0039*	*199.4790*
	Src 6	*200.0039*	*201.8520*	*200.0039*	*201.9780*
	Src 7	*200.0039*	*202.4190*	*200.0039*	*201.6840*
	Src 8	*199.9830*	*199.8779*	*199.9830*	*200.4660*
	Src 9	*200.0039*	*196.2030*	*200.0039*	*196.3920*

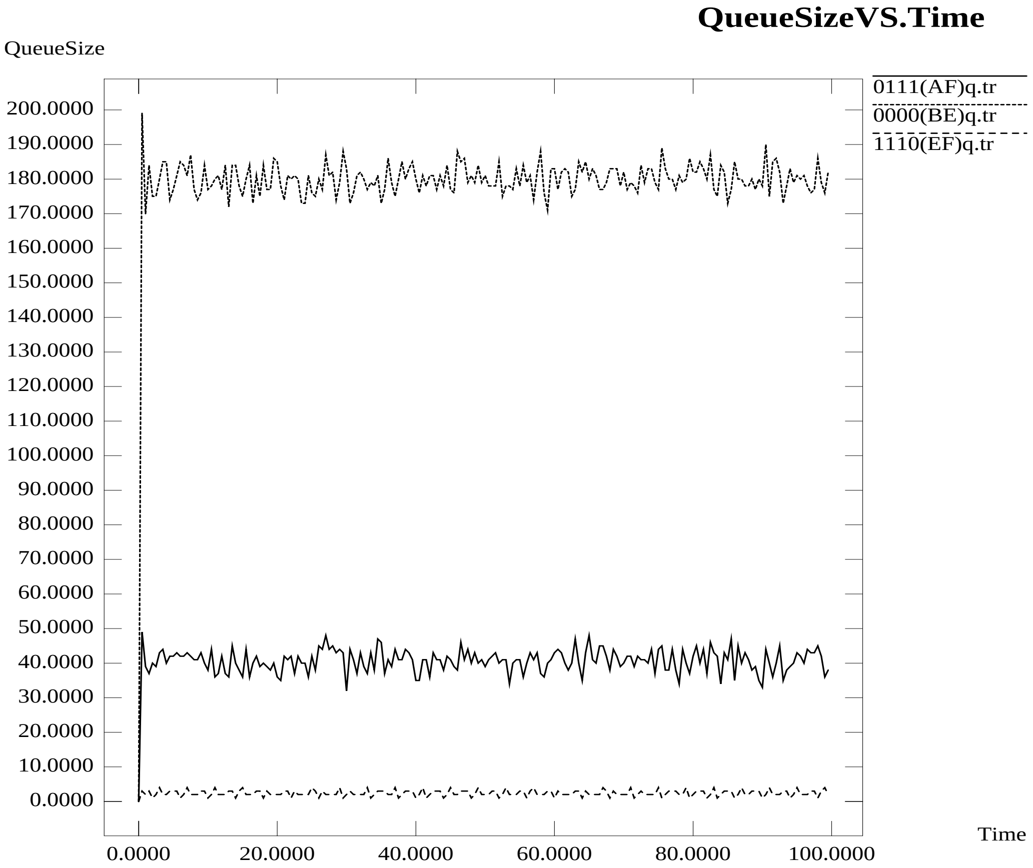
Drop Ratio of ATN Applications: Case 4

- ***Simulation results of Case 4:*** **Drop ratio of ATN traffic (measured at scheduler).**

Type of traffic	Case 1	Case 2	Case 3	Case 4
High Priority Traffic	*0.00000*	*0.00000*	*0.00000*	*0.00000*
Medium Priority Traffic	*0.00000*	*0.00000*	*0.49982*	*0.49982*
Normal Priority Traffic	*0.00000*	*0.66564*	*0.00000*	*0.66562*

- **Observations:** the drop ratio of both Medium and Normal Priority traffic are increased.

Queue Size Plot: Case 4



Queue Size Plot: Case 4

- **Observations:**
  - In this case, the high priority traffic has the smallest average queue size and jitter;
  - The normal priority traffic has the biggest average queue size and jitter.

Observations

- ***The high priority traffic*** always has the smallest jitter, the smallest average queue size and the smallest drop ratio without being affected by the performance of other traffic.

- ***The medium priority traffic*** has smaller drop ratio, jitter and average queue size than ***the normal priority traffic***.

Conclusion

- The high priority traffic receives the highest priority; the medium priority traffic receives higher priority than normal priority traffic.
- According to our simulation, the QoS requirements of ATN applications can be successfully achieved when ATN traffic is mapped to the DiffServ domain in the next generation Internet.

Task 3

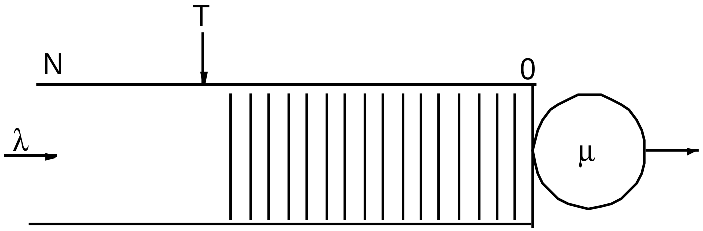
QoS in *DiffServ* over *ATM*

Prioritized EPD

- DS service classes can use the CLP bit of ATM cell header to provide service differentiation.
- EPD does not consider the priority of cells.
- Prioritized EPD can be used to provide service discrimination.
- Two thresholds are used to drop cells depending on the CLP bit.

Buffer Management Schemes

- EPD



•QL < T

Accept all packets.

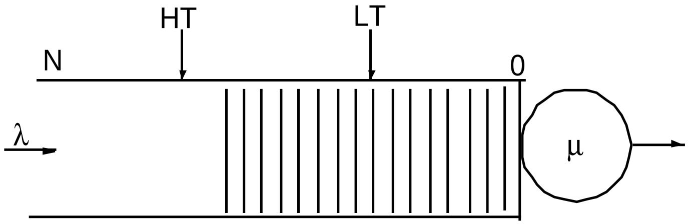
•T ≤ QL < N

Discard all new incoming packets.

•QL ≥ N

•Discard all.

- PEPD



•QL <LT

Accept all packets.

•LT ≤ QL < HT

Discard all new low priority packets.

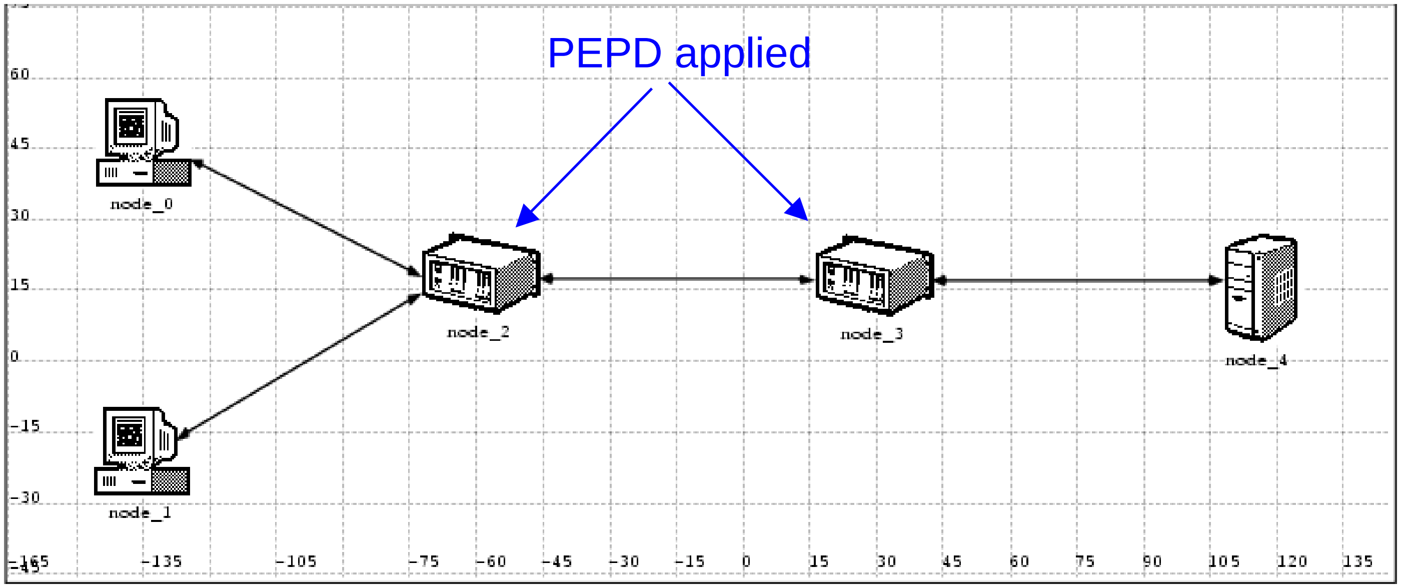
•HT ≤ QL < N

Discard all new packets

QL ≥ N

•Discard all packets

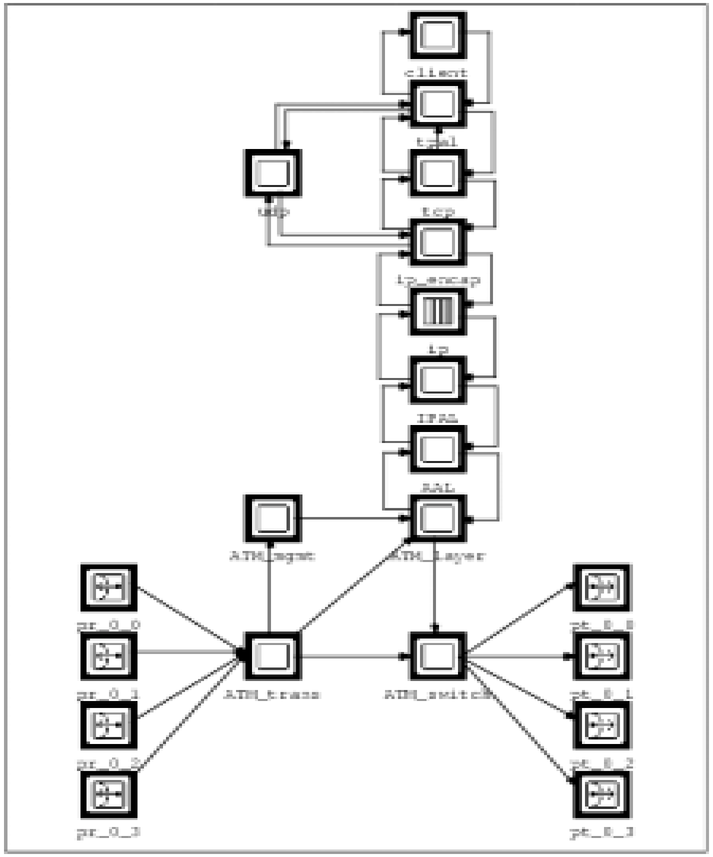
OPNET Simulation Configuration



PEPD applied

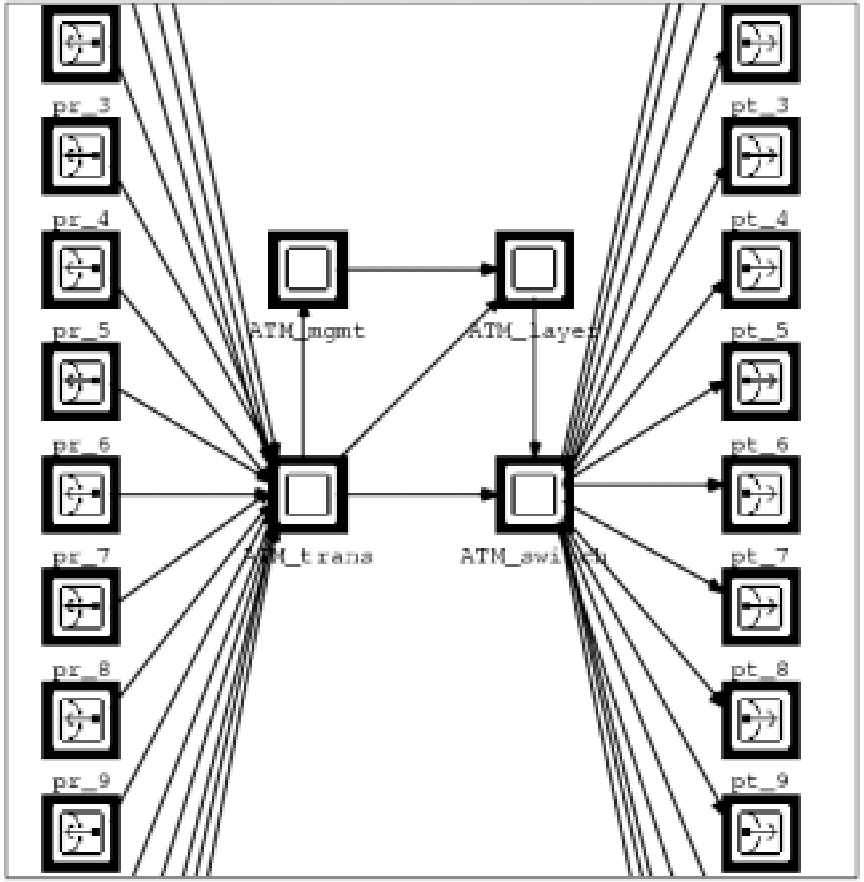
DS-ATM Protocol Stack

- AAL Layer marks the End of Packet.
- ATM_layer changes the CLP bit depending on the packet of the DS service.

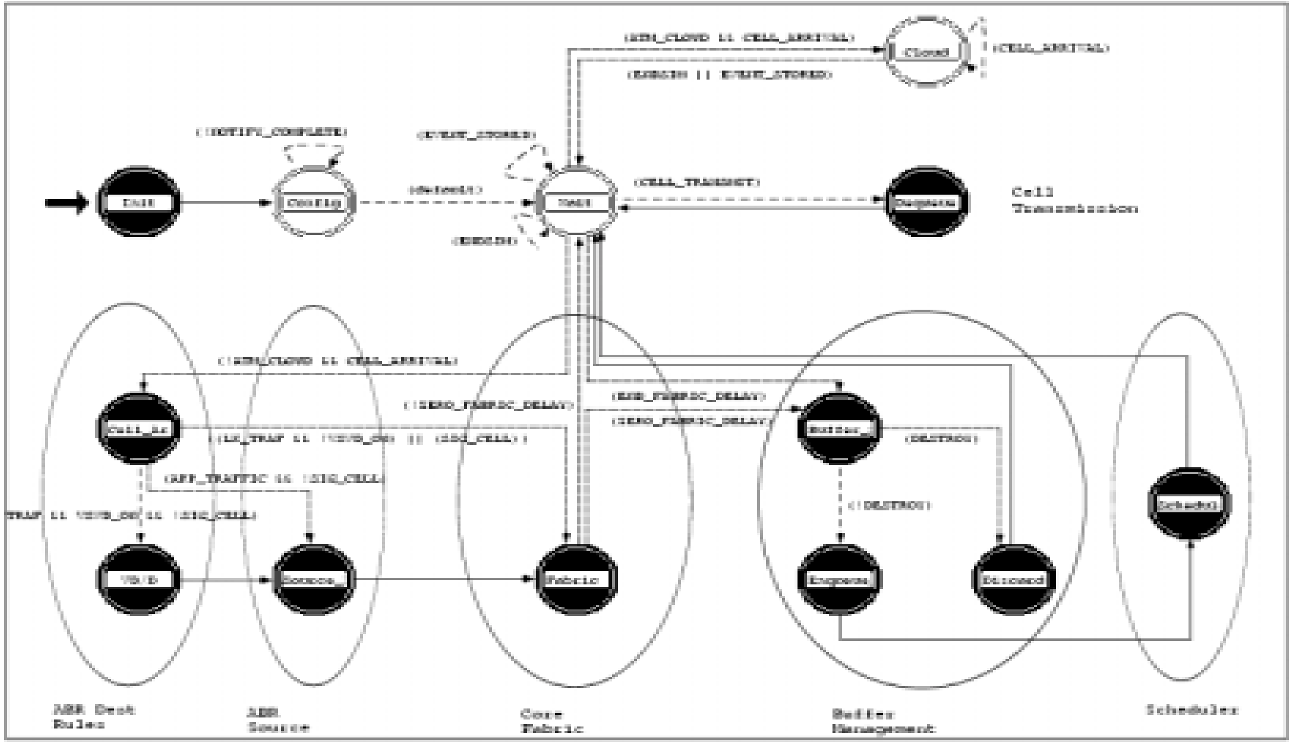


ATM Switch Node

- Support service differentiation in the ATM switch buffer.
- Change the buffer management scheme in the ATM_switch process to Prioritized EPD.

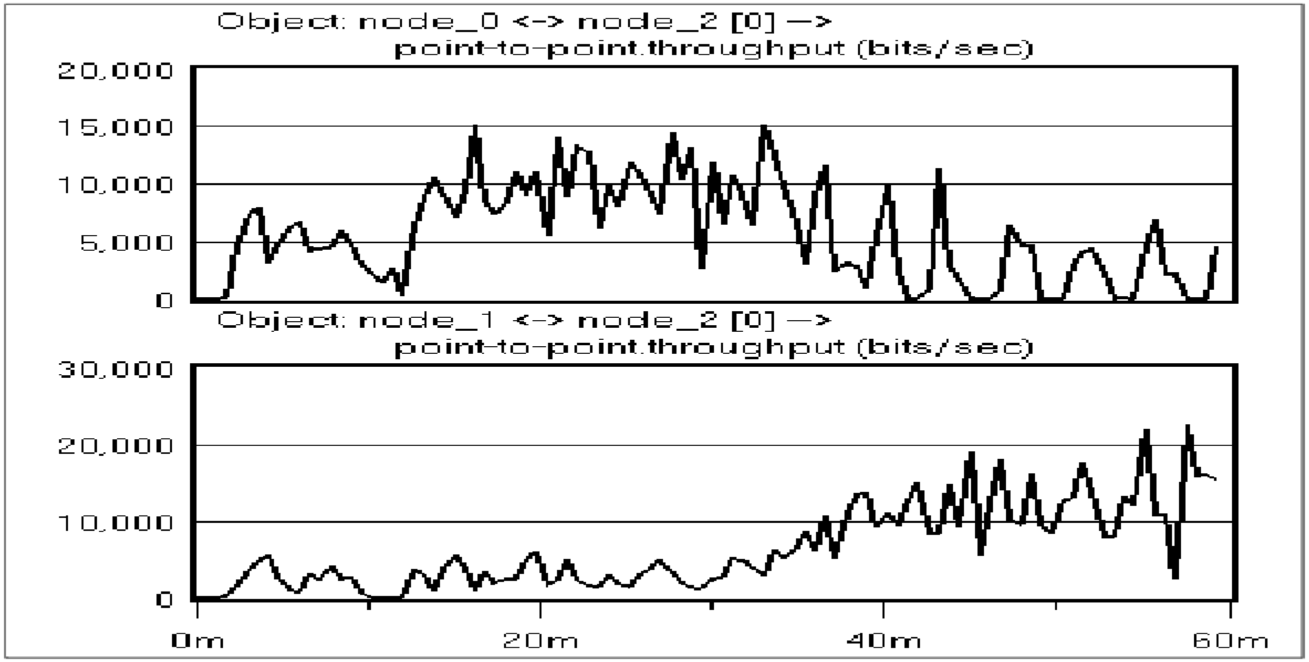


ATM_Switch Process

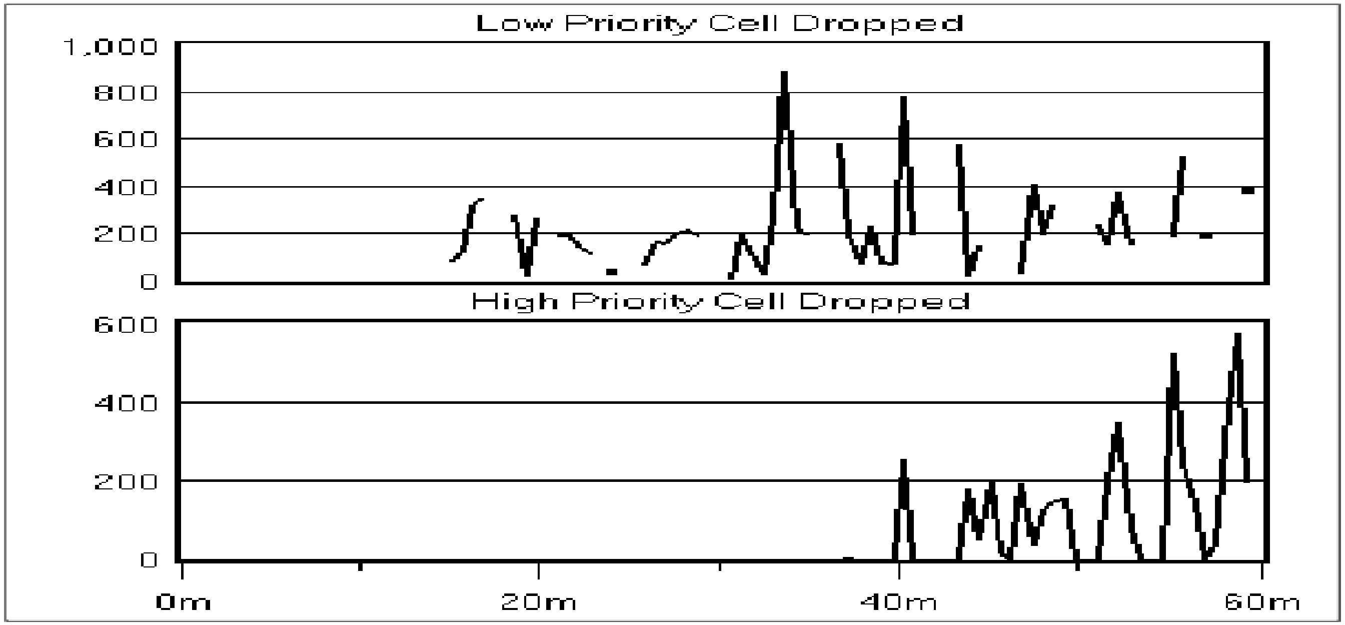


Implements the PEPD buffer management to support service differentiation.

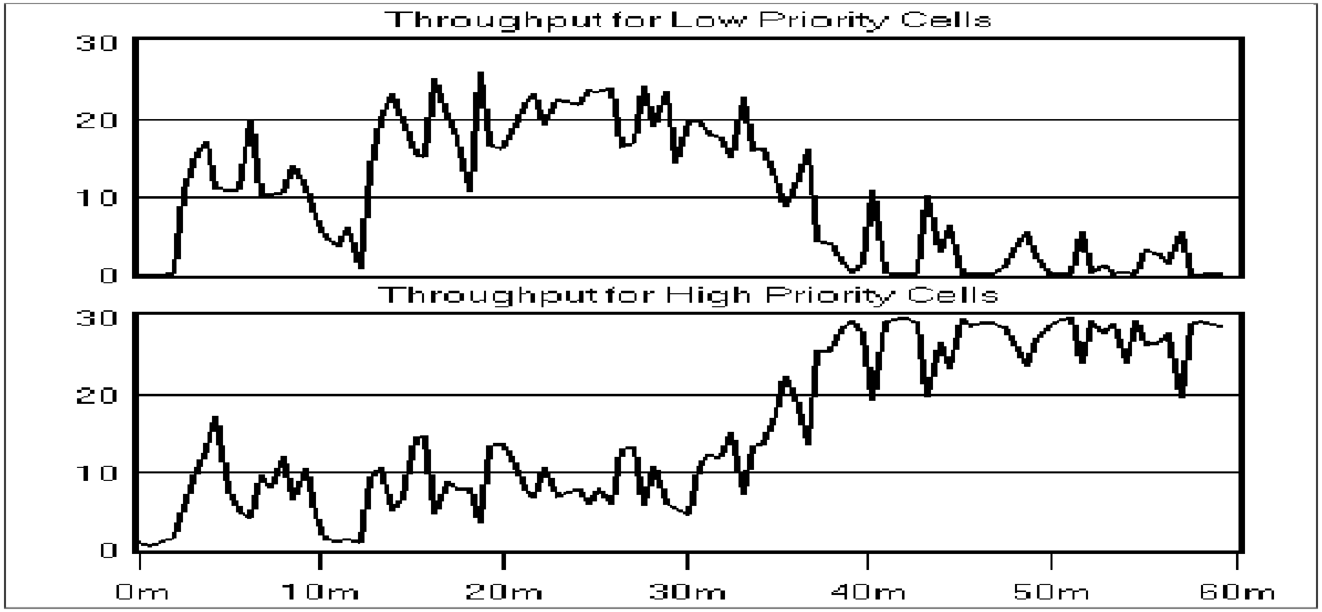
Source Rates



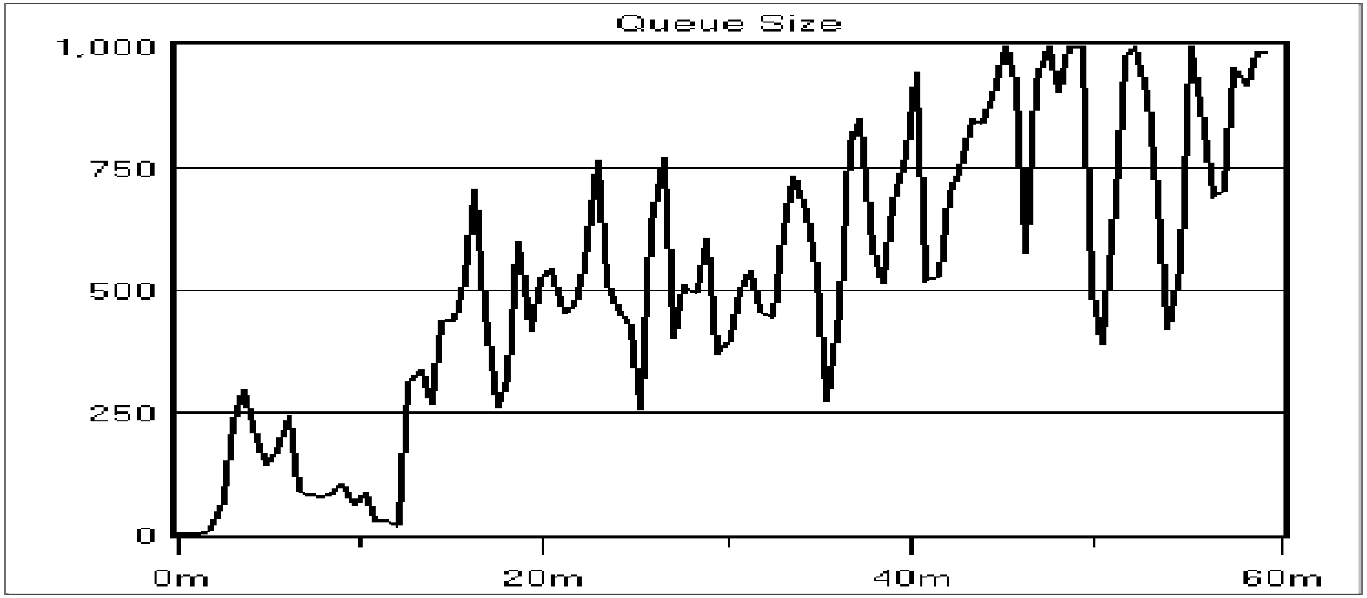
Cell Dropping



Throughput


Throughput for Low Priority Cells
30
20
10
0
Throughput for High Priority Cells
30
20
10
0
0m
10m
20m
30m
40m
50m
60m

Queue Occupancy



Conclusion

- Prioritized EPD can provide differential treatment to packets in an ATM core network.

- OVERALL: The tasks have been completed successfully.

- Thanks to NASA for the support of this project.

QoS for Real Time Applications over Next Generation Data Networks

Dr. Arjan Durresi

The Ohio State University
Columbus, OH 43210
durresi@cis.ohio-state.edu

These slides are available at

http://www.cis.ohio-state.edu/~durresi/talks/nasa1.htm

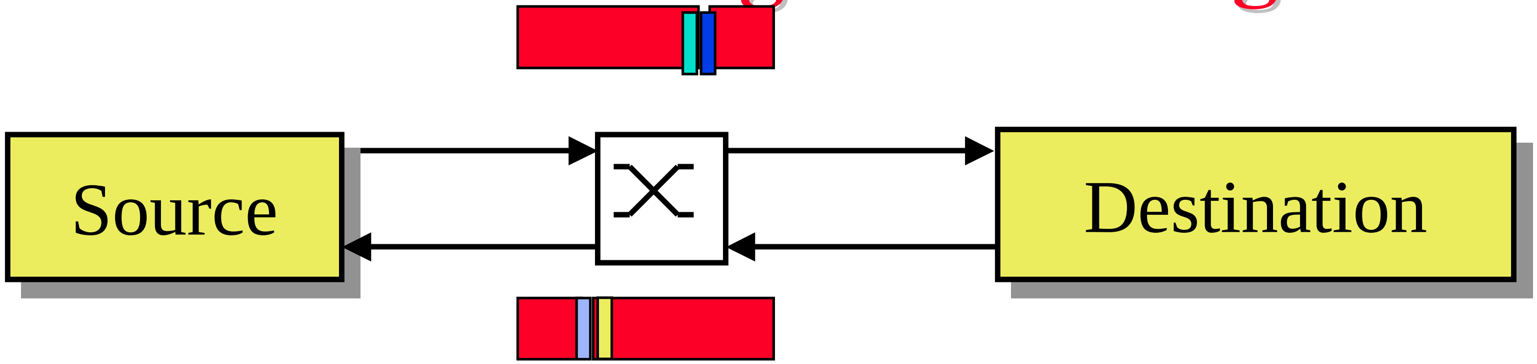
Overview

- □ Task 4: Improving Explicit Congestion Notification with the Mark-Front Strategy

- □ Task 5: Multiplexing VBR over VBR

- □ Task 6: Achieving QoS for TCP traffic in Satellite Networks with Differentiated Services

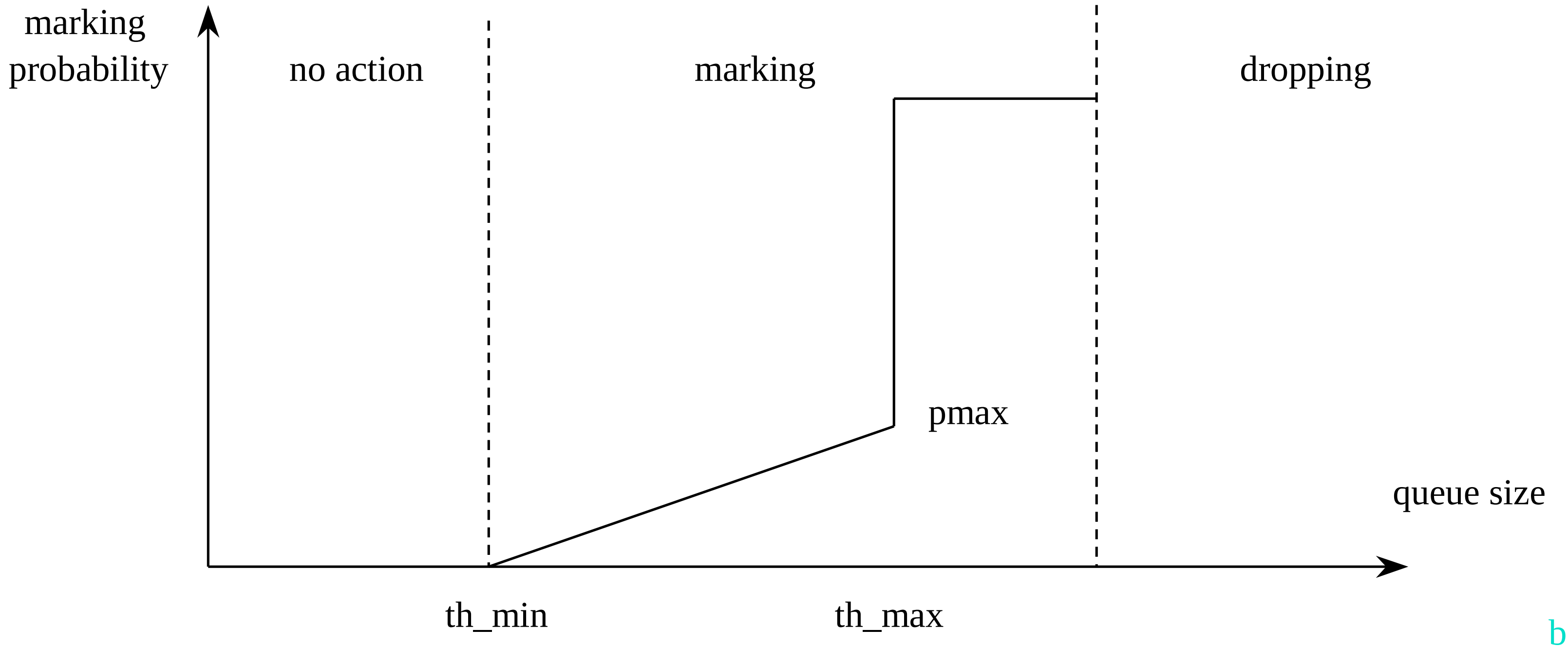
Task 4: Buffer Management using ECN



- Explicit Congestion Notification
- Standardized in RFC 2481, expected to be widely implemented soon.
- Two bits in IP header: Congestion Experienced, ECN Capable Transport
- Two bits in TCP header: ECN Echo, Congestion Window Reduced

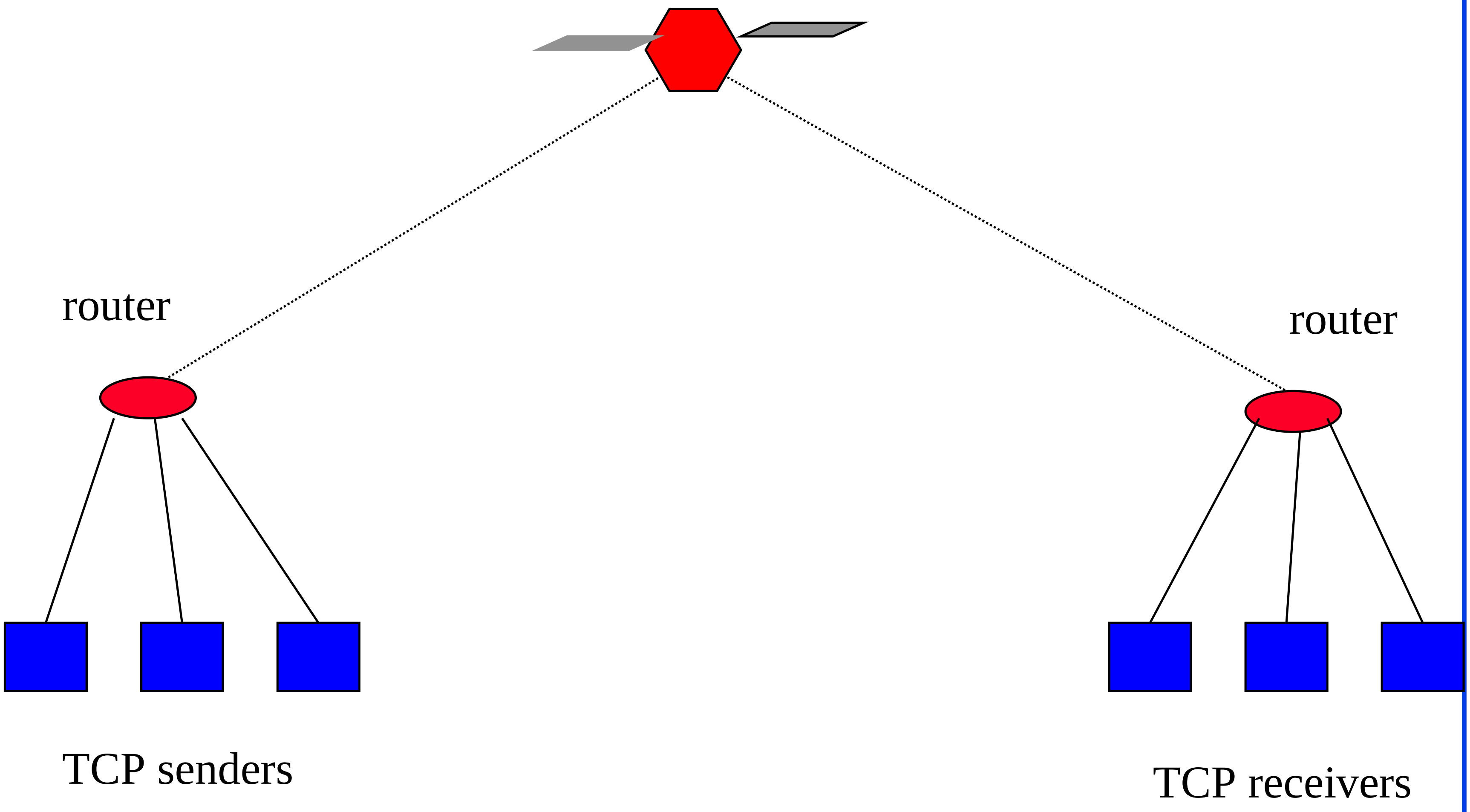
RED Marking and Dropping

- No standard action specified
- One possibility is to randomly mark at lower congestion levels. Random Early Detection (RED) marking instead of dropping.



back

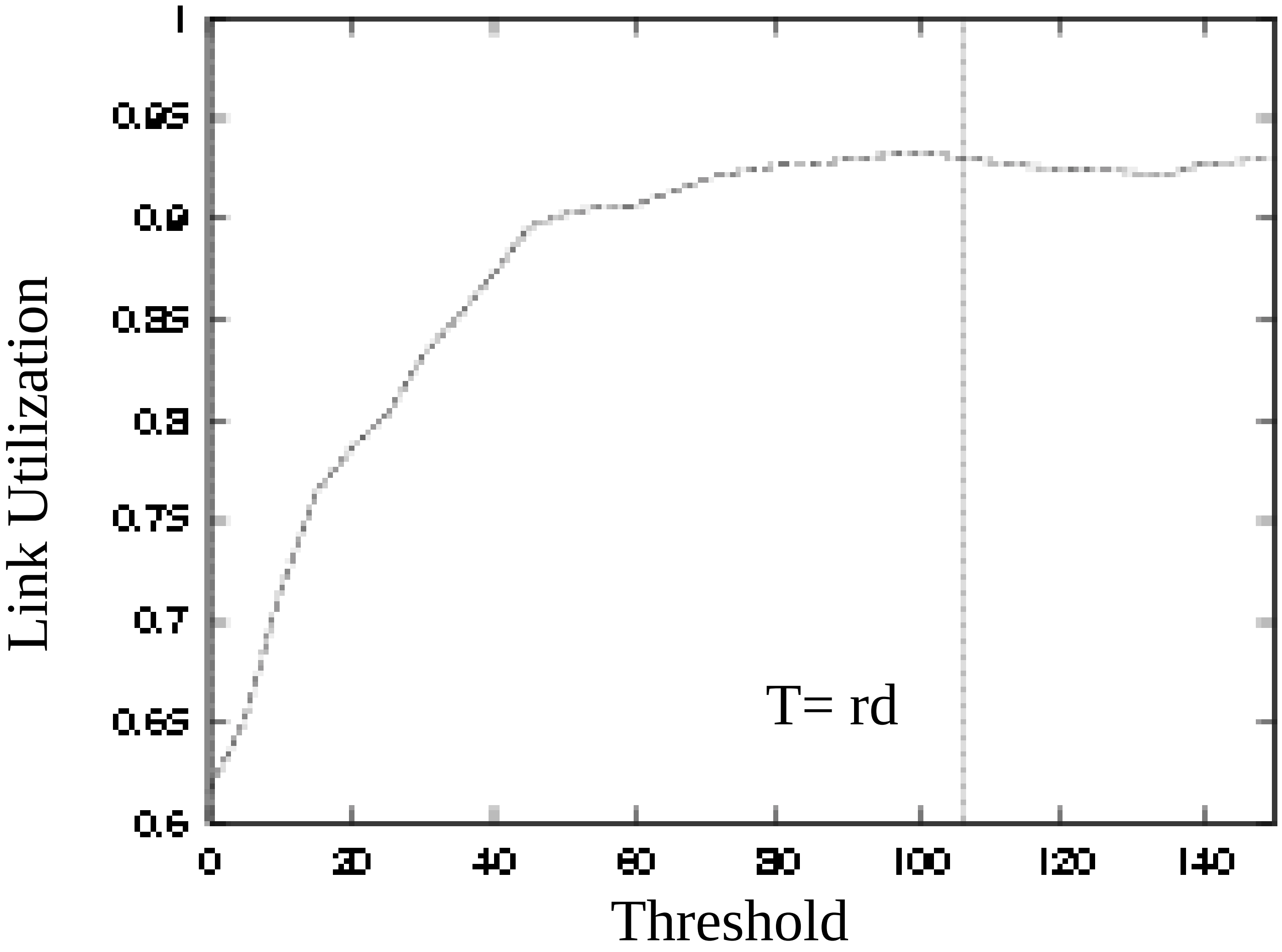
Simulation Model



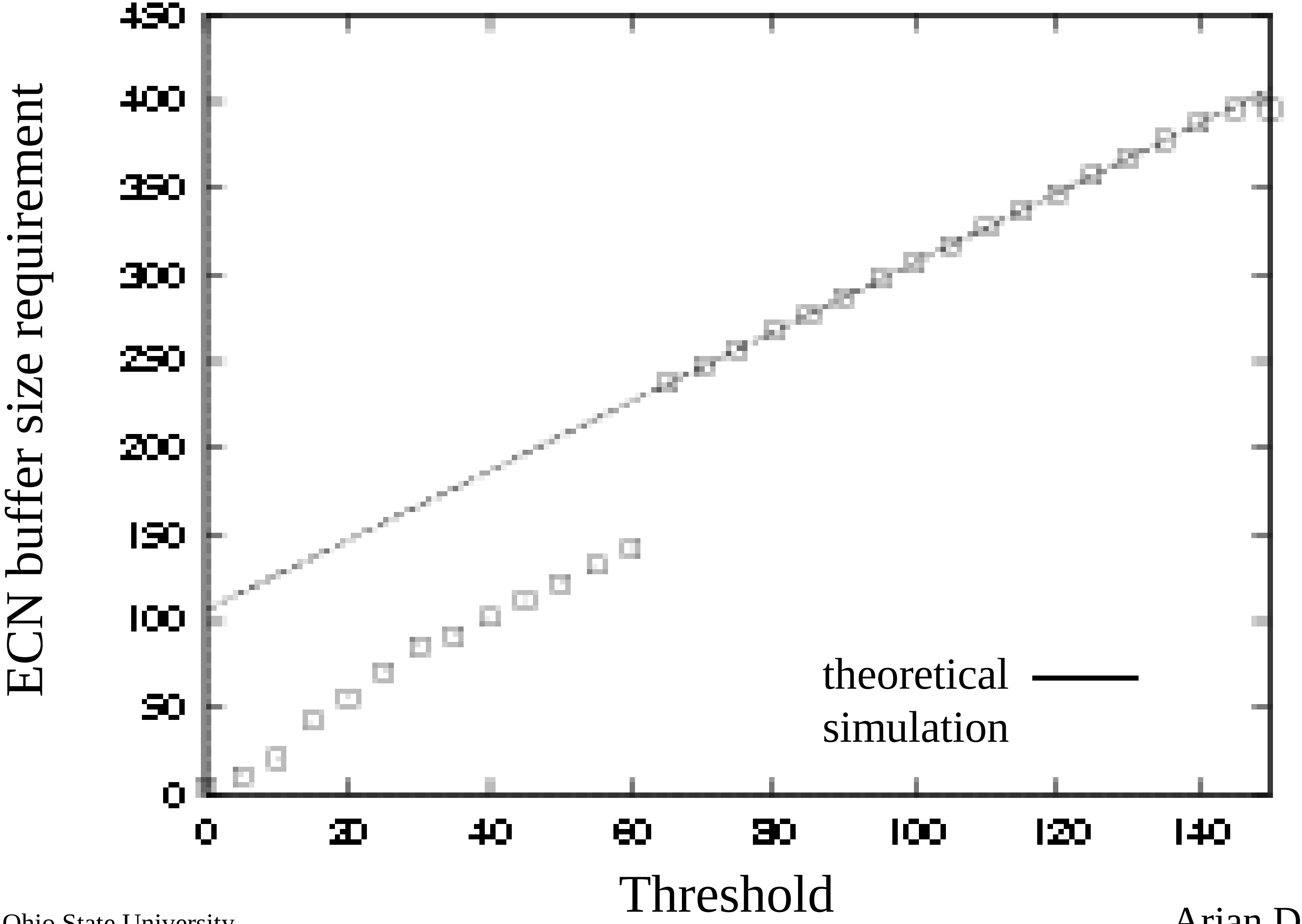
Threshold & Buffer Requirement

- In order to achieve full link utilization and zero packet loss, ECN routers should
  - have a buffer size of three times the bandwidth delay product;
  - set the threshold as 1/3 of the buffer size;
- Any smaller buffer size will result in packet loss.
- Any smaller threshold will result in link idling.
- Any larger threshold will result in unnecessary delay.

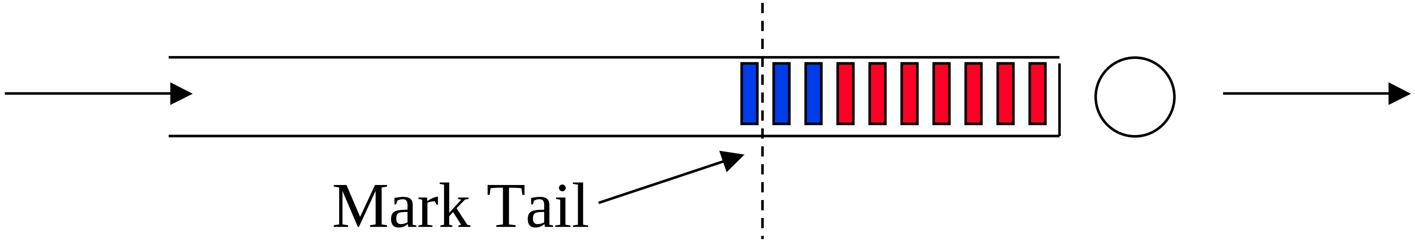
Setting the Threshold



Buffer Requirement for No Loss

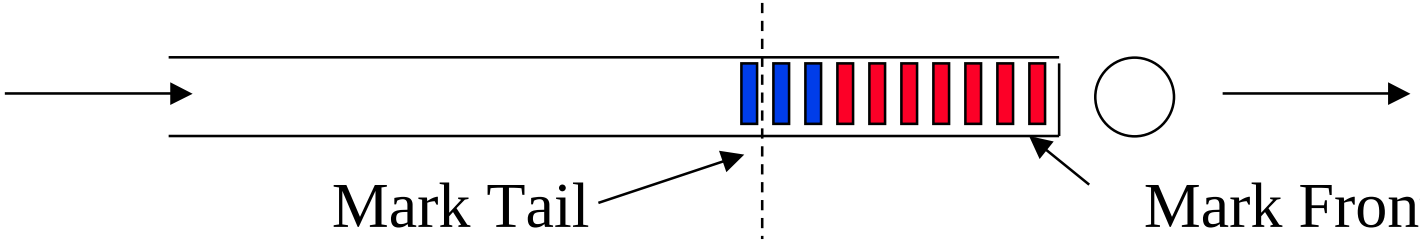


Problem with Mark Tail



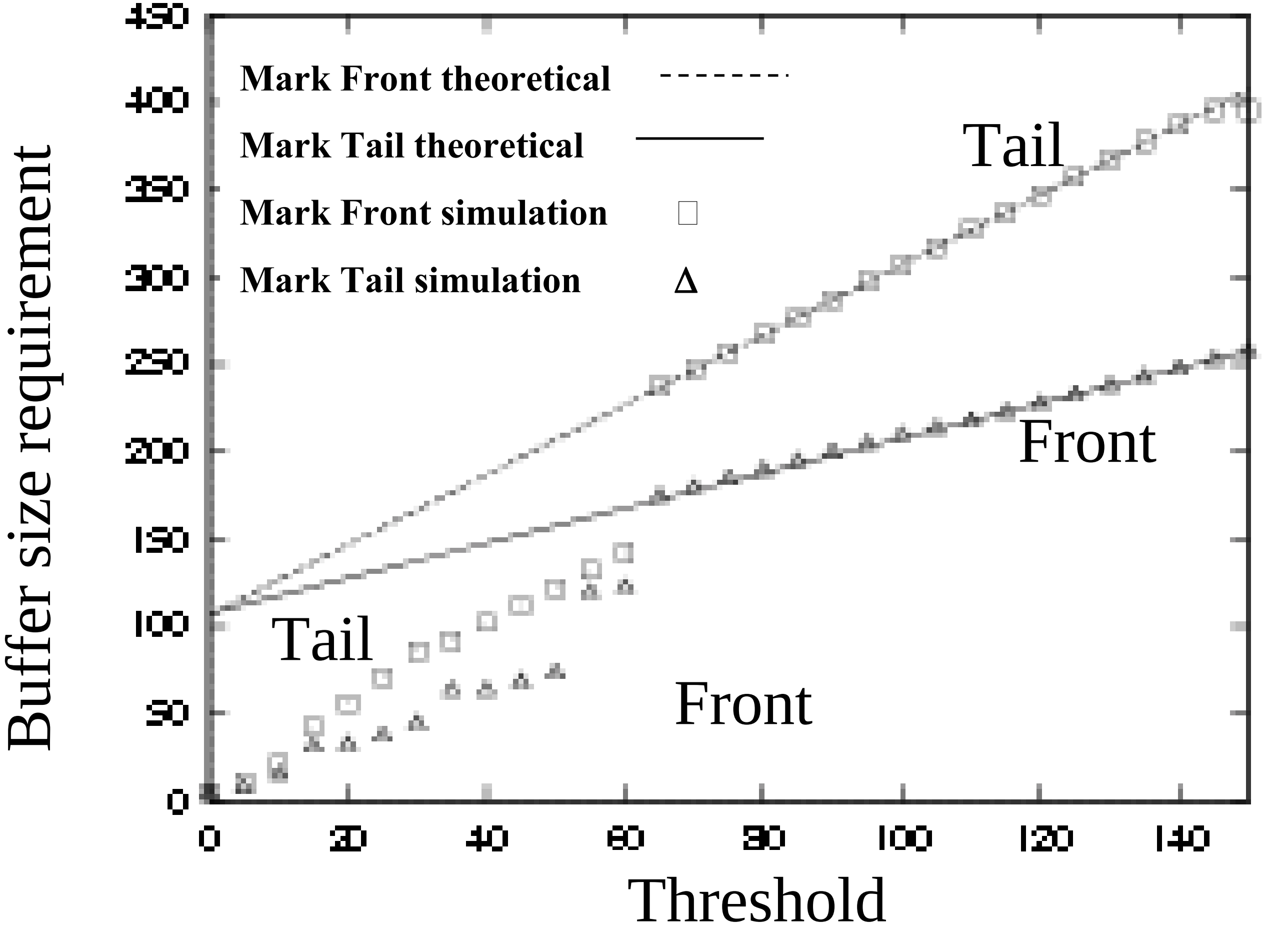
- Current ECN marks packets at the tail of the queue:
  - congestion signals suffer the same delay as data;
  - when congestion is more severe, the ECN signal is less useful;
  - Flows arriving when buffers are empty may get more throughput ⇒ unfairness.

Proposal: Mark Front



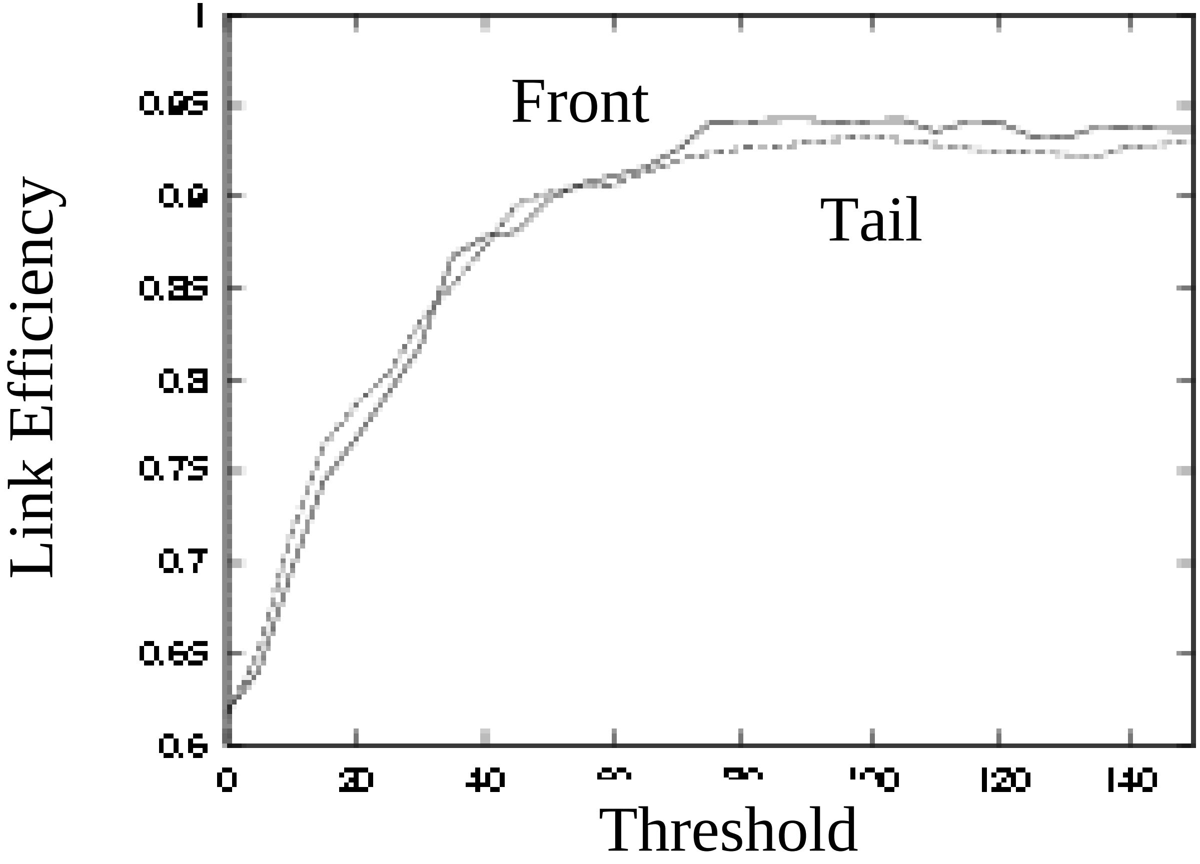
- Simulation Analysis has shown that mark-front strategy
  - reduces the buffer size requirement, from 3rd to 2rd
  - increases link efficiency,
  - avoids lock-in phenomenon and improves fairness,
  - alleviates TCP’s discrimination again large RTTs.

Buffer Requirement



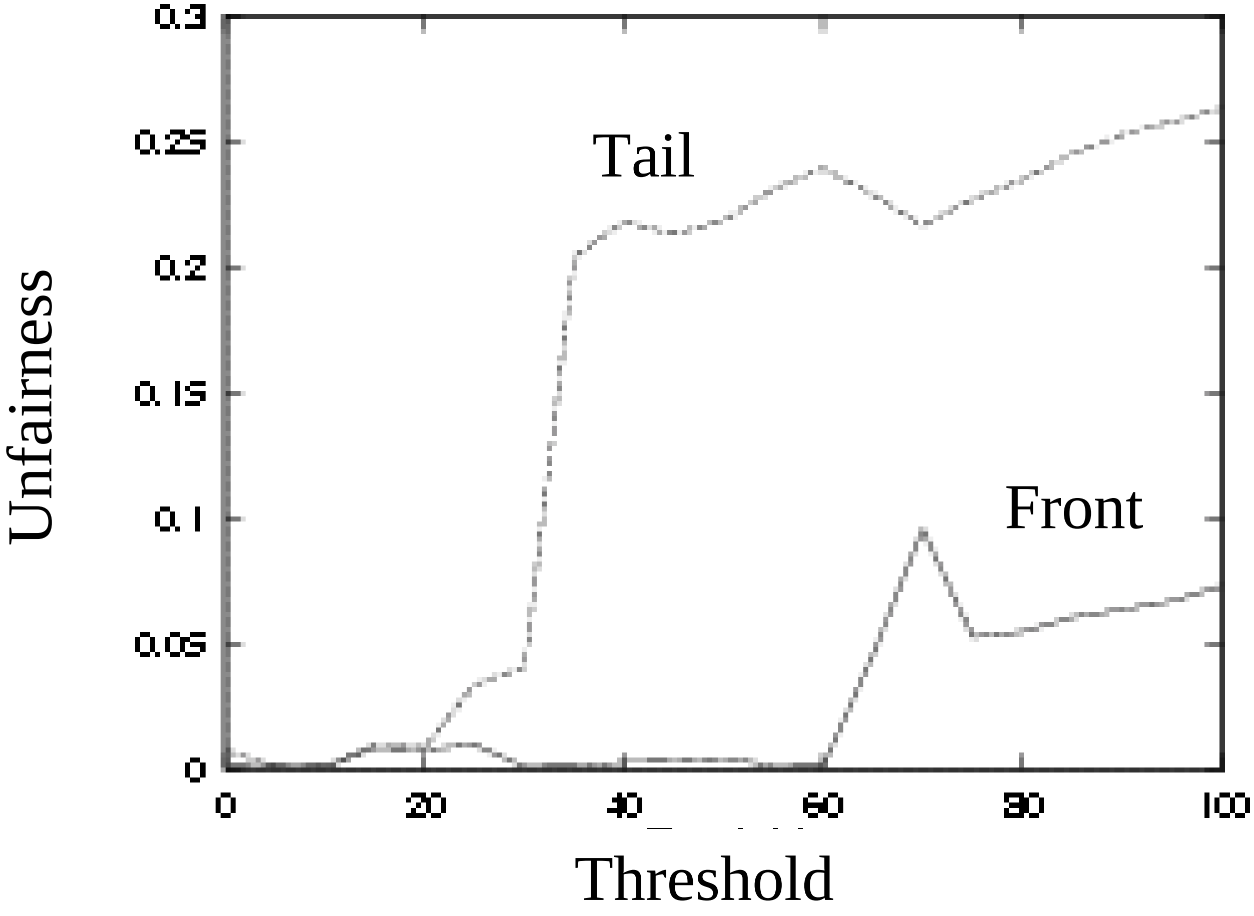
- Our theoretical bound is tight when threshold ≥ rd
- Mark Front requires less buffer

Link Efficiency



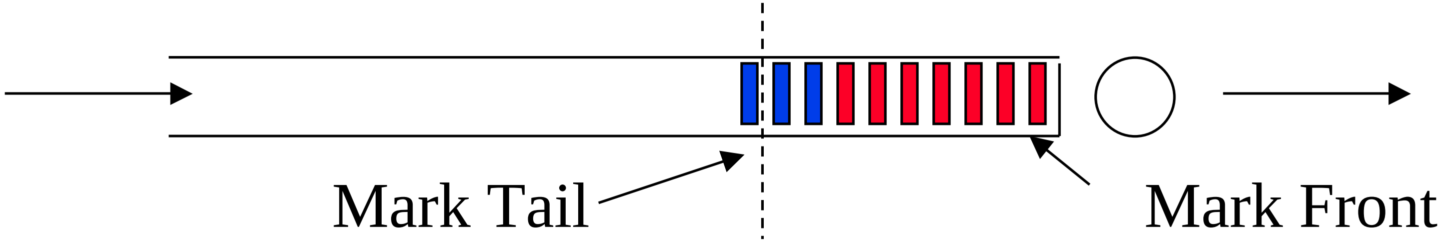
- ❑ Mark-Front improves the link efficiency for the same threshold

Unfairness



- Mark-Front improves fairness

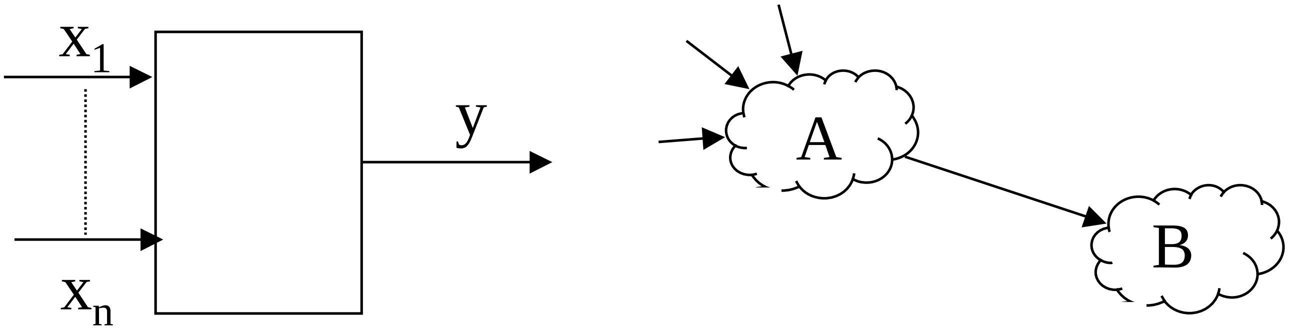
Task 4: Summary of Results



- Mark-front strategy
  - reduces the buffer size requirement,
  - increases link efficiency,
  - avoids lock-in phenomenon and improves fairness
- This is specially important for long-bandwidth delay product networks

Task 5: Multiplexing

- Internet is becoming the new infrastructure for telecommunications.
- Everything over IP and IP over everything
- IP has to provide one key function: Decomposition of high-capacity channels into hierarchically ordered sub-channels
- Or how to build a multiplexing node?



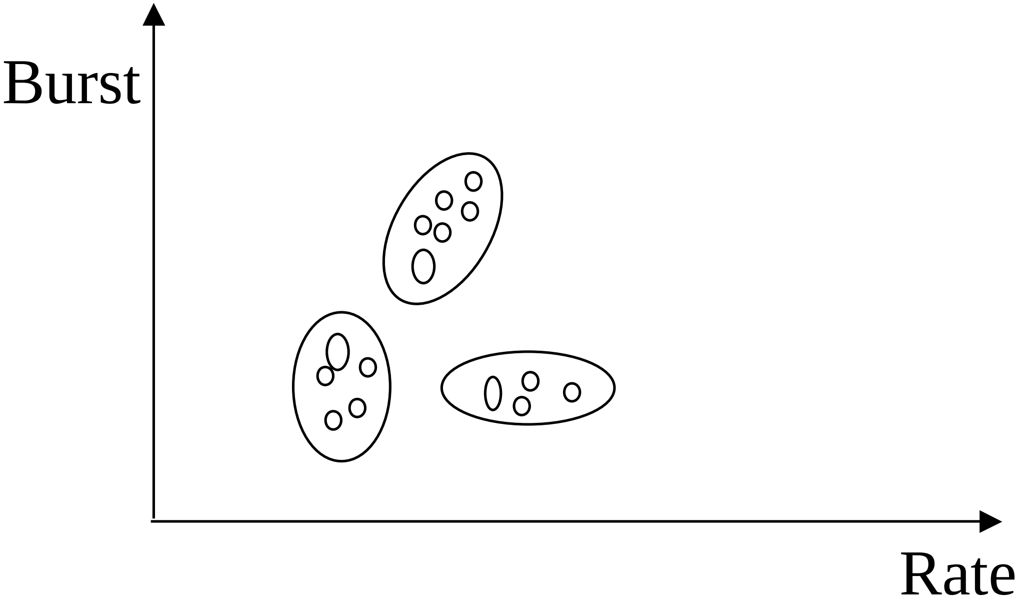
Multiplexing (cont.)

- ❑ Real-time or high QoS traffics include voice over IP, virtual leased line, video over IP etc.
- ❑ Goal and Objectives:
  - ❑ Provide the needed QoS
  - ❑ High network resource utilization or multiplexing gain
  - ❑ “Simple” solutions to be deployed extensively in practice

Problems

- How to characterize IP traffic to be aggregated:
  - Different models and parameters
  - Models should be “practical” i.e. easy to be used by applications
- How to characterize the output traffic
- Provide multiplexing rules: balancing QoS for the aggregates and multiplexing gain: find an optimized solutions
- Select the appropriate scheduling mechanism

Optimized Solution



- Find the best rules how to multiplex different IP traffics

How to characterize IP traffic flows

- Envelope-Based methods to specify the traffic, both deterministically or stochastically
- Leaky Bucket Model
  - Using Leaky Bucket Model to do simulation
- D-BIND/H-BIND Model

Envelope Concepts

- $E(t) >= A[s, s+t]$, for all $t > 0$, and $s > 0$
- Empirical Envelope is the tightest envelope for a given traffic
- Definition: Let A(t) be the arriving traffic, then

  $$E(t) = \max_{s>0} A[s, s+t], \text{ for all } t > 0$$

  is the Empirical Envelope of A(t). A[t1, t2] represents the amount of traffic arrives during interval [t1, t2]
- Mathematical tool: Network Calculus to do envelope based derivation and analysis

Leaky Bucket Model

- Most research work uses this model to characterize the arriving traffic
- $E(t) = b + r*t$, b is the bucket size, and r is the leaking rate
- $E(t) >= A[s, s+t]$, where $A(t)$ is the arriving traffic
- Multiple Leaky Bucket Model
- $E^*(t) = \min_{1<=I<=n}\{b_i + r_i*t\}$

D-BIND/H-BIND Model

- Proposed by Edward Knightly and Hui Zhang
- With the D-BIND model, sources characterize their traffic to the network via multiple rate-interval pairs,(R_k, I_k), where the rate R_k, is a bounding or worst-case rate over every interval of length I_k. With P rate-interval pairs, the model parameterizes a piece-wise linear constraint function with P linear segments given by

  $E(t) = (R_{k*}I_k - R_{k-1}*I_{k-1})(t-I_k)/(I_k - I_{k-1}) + R_k*I_k$, $I_{k-1} < t <= I_k$ with $I_0 = 0$;

D-BIND/H-BIND Model (Cont.)

- Simulations have showed that, P, the number of pairs needs not to be too large. P=4 is good.
- Better performance than Leaky Bucket Model
- Better describe the correlation structure and burstiness properties for a given traffic(Video, etc)
- Drawbacks:
  - Larger number of parameters to characterize the traffic
  - Unrealistic to let users to accurately specify such parameters, need some on-line traffic measurement

D-BIND/H-BIND Model (cont.)

- Multiple Leaky Bucket Model is a special case of D-BIND
- H-BIND extends D-BIND
- It uses D-BIND to characterize traffic, and achieves statistical multiplexing

How to get characteristics of aggregates of several IP flows

- Deterministic Approaches
- Stocastic/Statistical Approaches

Deterministic Approaches

- Consider worst case
- Provide 100% QoS guarantees
- Drawback:
  - waste network resource
  - bandwidth utilization is low(under 50% with D-BIND)

Stocastic/Statistical Approaches

- No 100% guarantee
- Probabilistic guarantee. For example, to guarantee packet loss rate is smaller than 10^{-6}.
- Better network utilization.

Stocastic/Statistical Approaches (Cont.)

- Connection Admission Control Algorithms for statistical QoS.
  - Average/Peak Rate combinatorics
  - Additive Effective Bandwidths
  - Loss Curve
  - Maximum Variance Approaches
  - Refined Effective Bandwidths and Large Deviations
  - Measurement based algorithms.
  - Enforceable statistical service
  - Algorithms for special-purpose system(video on demand)

CBR vs. VBR

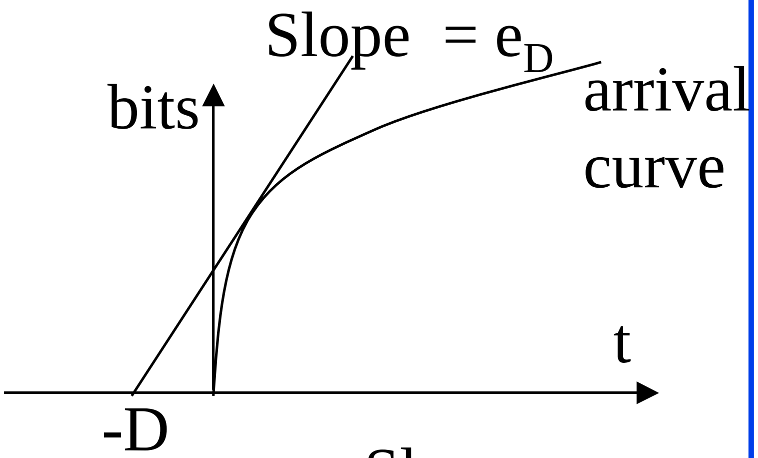
- IP CBR: Simple multiplexing rules, Provide guarantee for QoS, No multiplexing gain
- IP VBR, More complex multiplexing rules, Guarantees for QoS depend on multiplexing rules, More multiplexing gain:
  - With CBR overbooking is performed by burst absorption at the multiplexer.
  - With VBR it is possible to let burst go through the multiplexer and count on statistical multiplexing inside the network, where the number of connections and the trunk bit rates are larger

Output - CBR

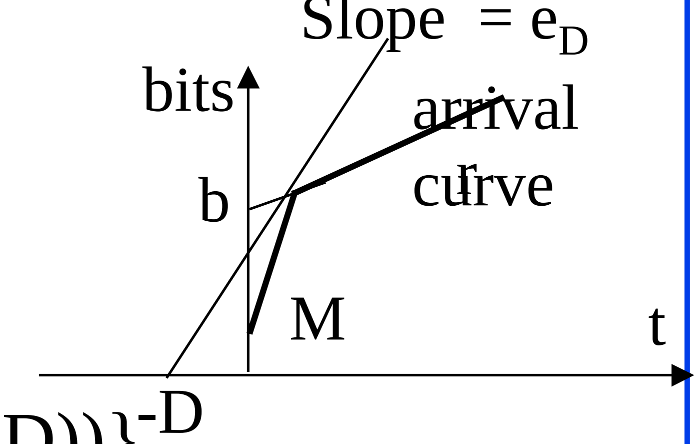
- Effective Bandwidth:

The queue with constant rate C guarantees a delay bound D to a flow with arrival curve α if $C \geq e_D(\alpha)$ where:

$e_D(\alpha) = \sup \alpha(s)/(s+D)$ for $s \geq 0$



- Example: For IETF traffic specification:

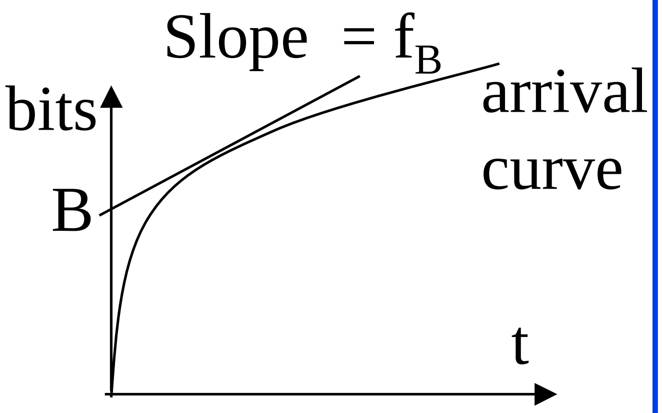


$e_D=\max\{M/D, r, p, (1-(D-M/p)/(x+D))\}$

where $x = (b-M)/(p-r)$

- $\sum e_D(\alpha_i) - e_D(\sum \alpha_i)$ is non statistical multiplexing gain

Output - CBR (cont.)



- Equivalent capacity

Bound the buffer B:

$C \geq f_B(\alpha) = \sup (\alpha(s)-B)/s;\ s \geq 0$

- Again: $f_B(\alpha) \leq \sum f_{Bi}(\alpha_i)$

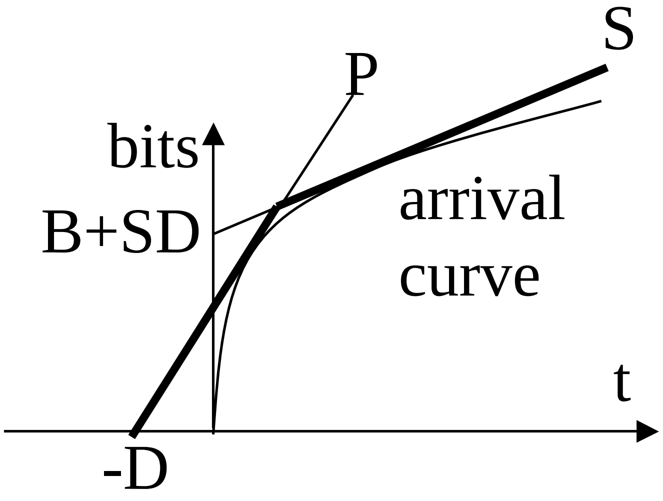
- Also given a predicted traffic we can find the optimal VBR parameters that can carry the traffic.

Output - VBR

- For $\sigma(t) = \min(Pt, St+B)$

P: Peak rate, S: sustainable rate,

B: burst tolerance



$$\begin{cases} (1)\ (s+D)P \geq \alpha(s) \\ (2)\ (s+D)S+B \geq \alpha(s) \end{cases}$$

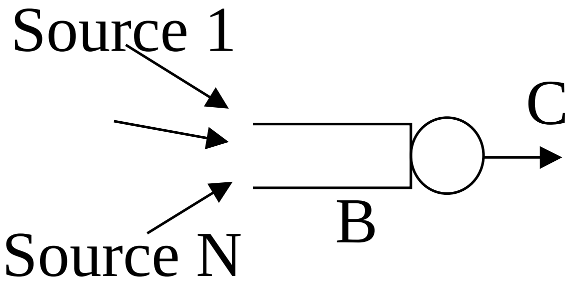
- From (1)=> $P \geq P_0 = e_D(\alpha)$
- Using a VBR trunk rather than a CBR is all benefit since, by definition of effective bandwidth, the CBR has at least a rate P_0
- We can also optimize S and B.

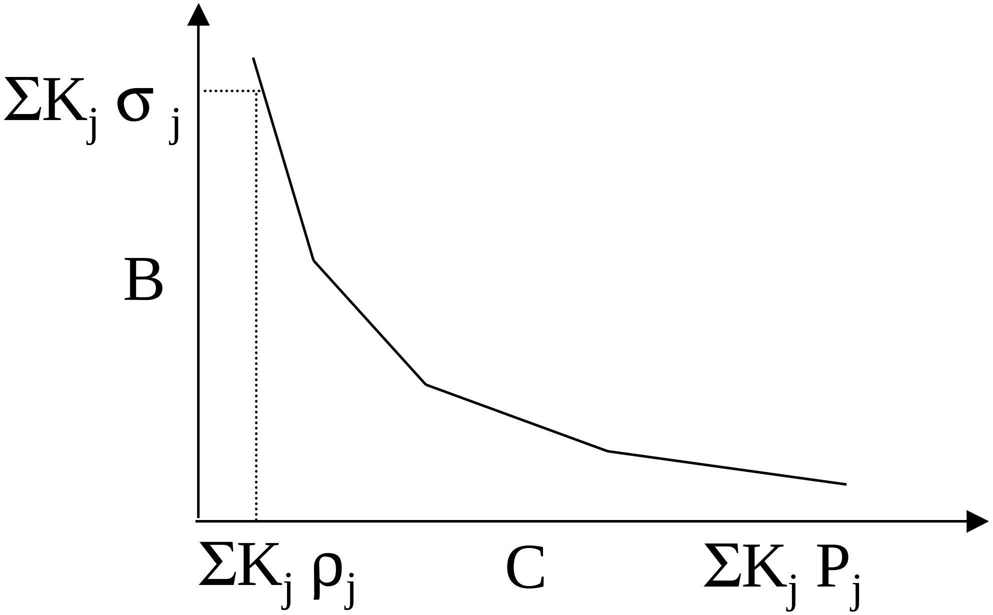
Output - VBR

- Given a predicted traffic we can find the optimal VBR parameters that can carry the traffic.
- A number of flows, with an aggregated curve α, is multiplexed into a VBR trunk
- The VBR trunk is viewed as a single stream by downstream nodes and is constrained by an arrival curve σ.
- If the constraint at the multiplexor is to guarantee a maximum delay D:
  - $\sigma(s + D) \geq \alpha(s)$ for $s \geq 0$

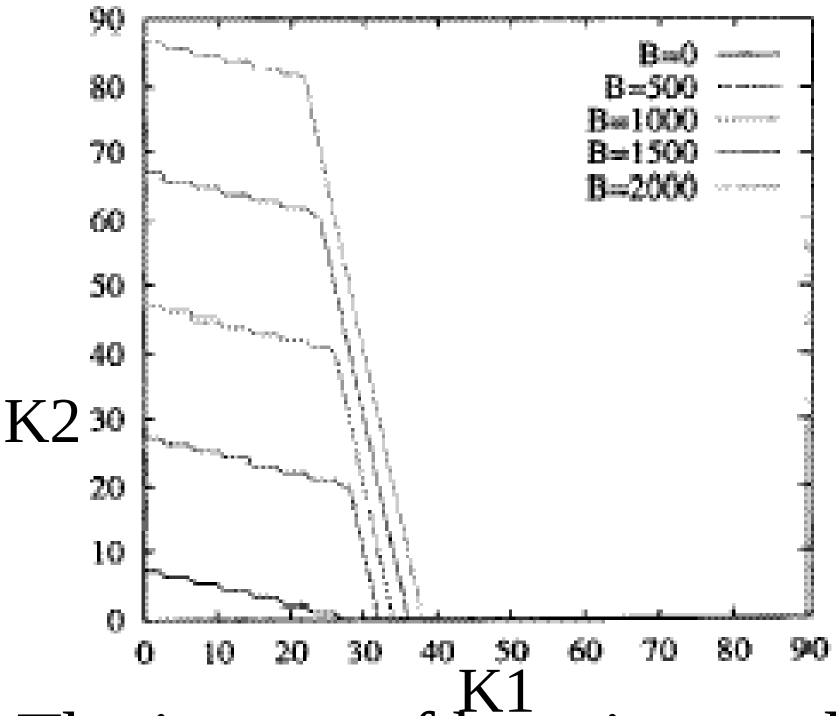
CBR Multiplexor

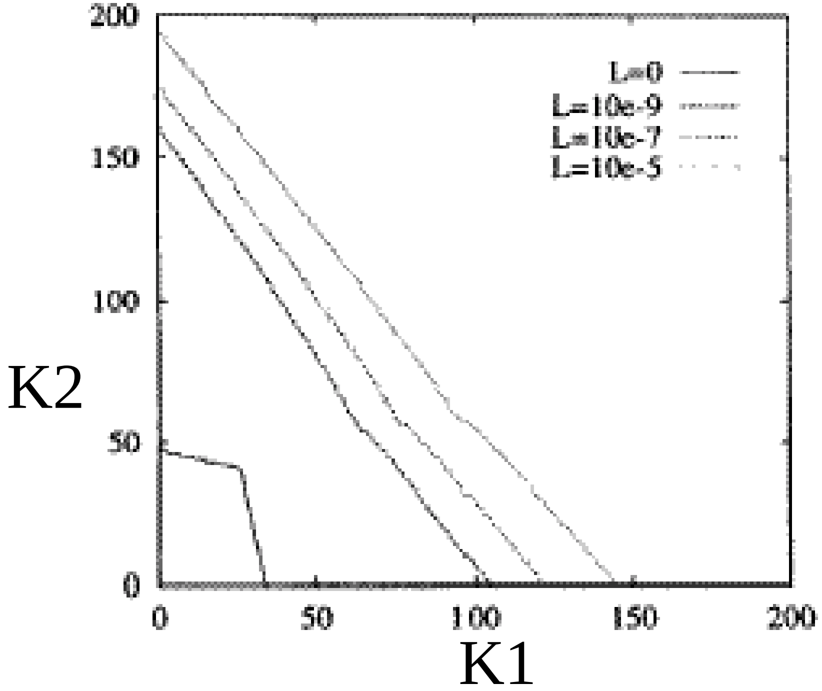
- Each source is leaky bucket regulated: token rate ρ, token size σ and the peak rate P
- We can calculate queue length.
- We can build the optimal Buffer/Bandwidth curve





CBR Multiplexor



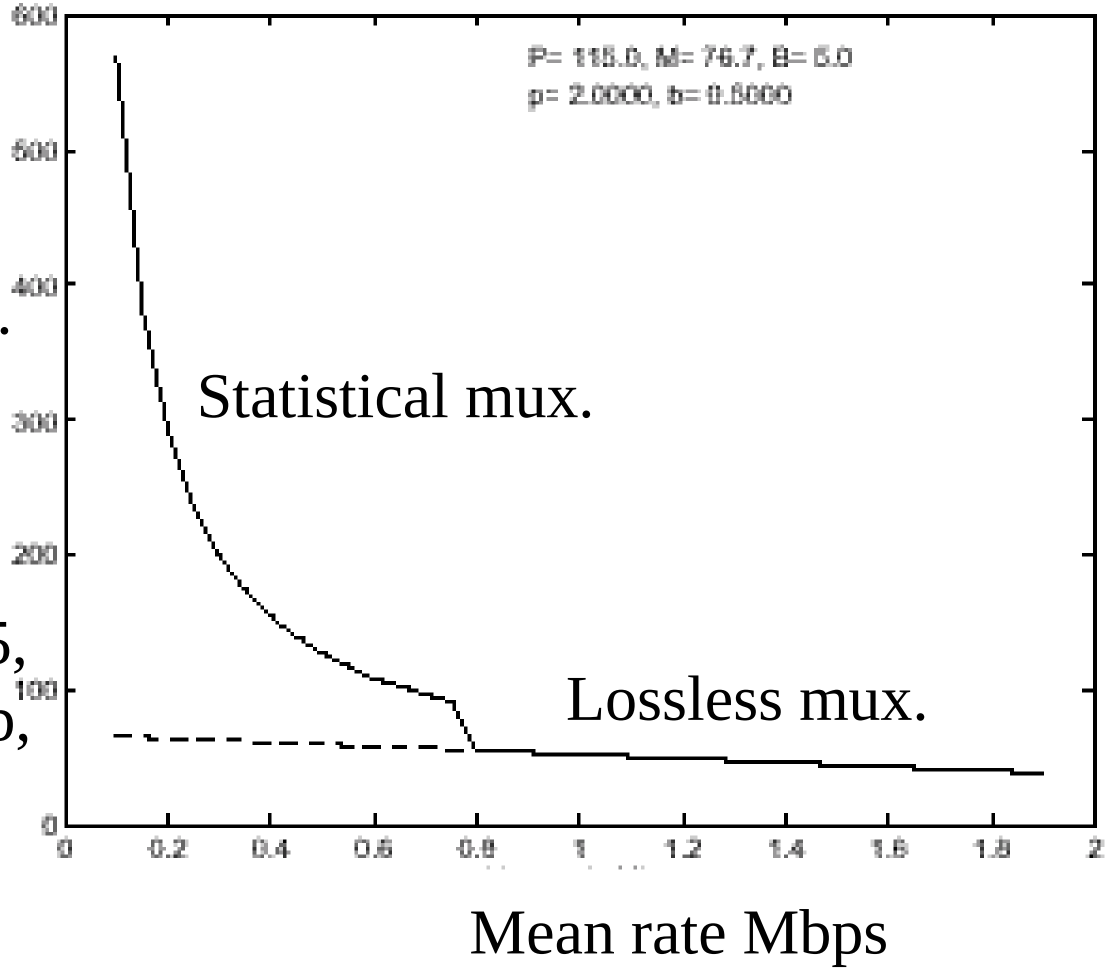


- The impact of burstiness when heterogeneous sources are multiplexed C=45Mbps, $\rho_1 = \rho_2 = 0.15$, $P_1=1.5$, $P_2=6$, $T_{on1} >> T_{on2}$
- Statistical service with even small loss probability increases the multiplexing gain.

VBR Multiplexor

- Advantages of using VBR trunks:
  - More statistical multiplexing gain than CBR
  - With CBR overbooking is performed by burst absorption at the multiplexer.
  - With VBR it is possible to let burst go through the multiplexer and count on statistical multiplexing inside the network, where the number of connections and the trunk bit rates are larger.

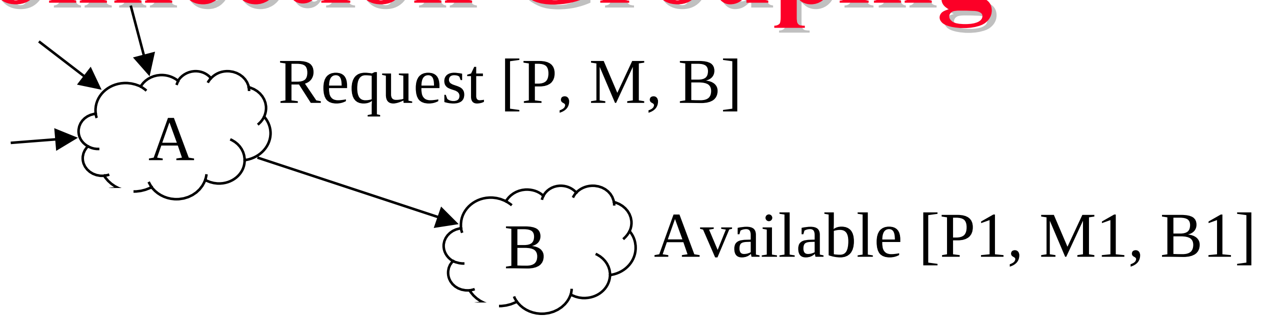
VBR Multiplexor

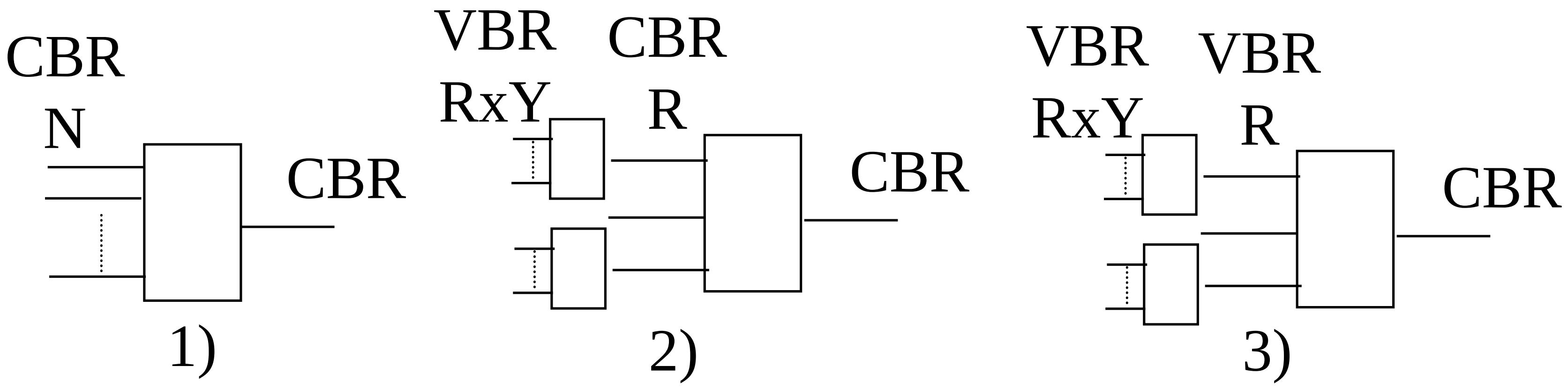


Loss prob. =10 $^{-9}$,
Peak=115Mbps,
M (sustained)=Peak/1.5,
B (bust tolerance)=5Mb,
X(buffer)=10Mb
Input VBR flows:
p=2 Mbps, b=0.5Mb
m = 0 to p

Figure shows the advantage of statistical multiplexing

Connection Grouping





- Compare three type of multiplexing:
  - 1) No connection grouping
  - 2) Connections grouped as CBR trunks
  - 3) Connections grouped as VBR trunks

VBR Multiplexor

1) and second stage of

2) and 3):

P = 115Mbps N. conn.

X=10Mb

Smaller node in 2):

P=11.5Mbps

X=1Mb

Smaller node in 3):

P=11.5Mbps

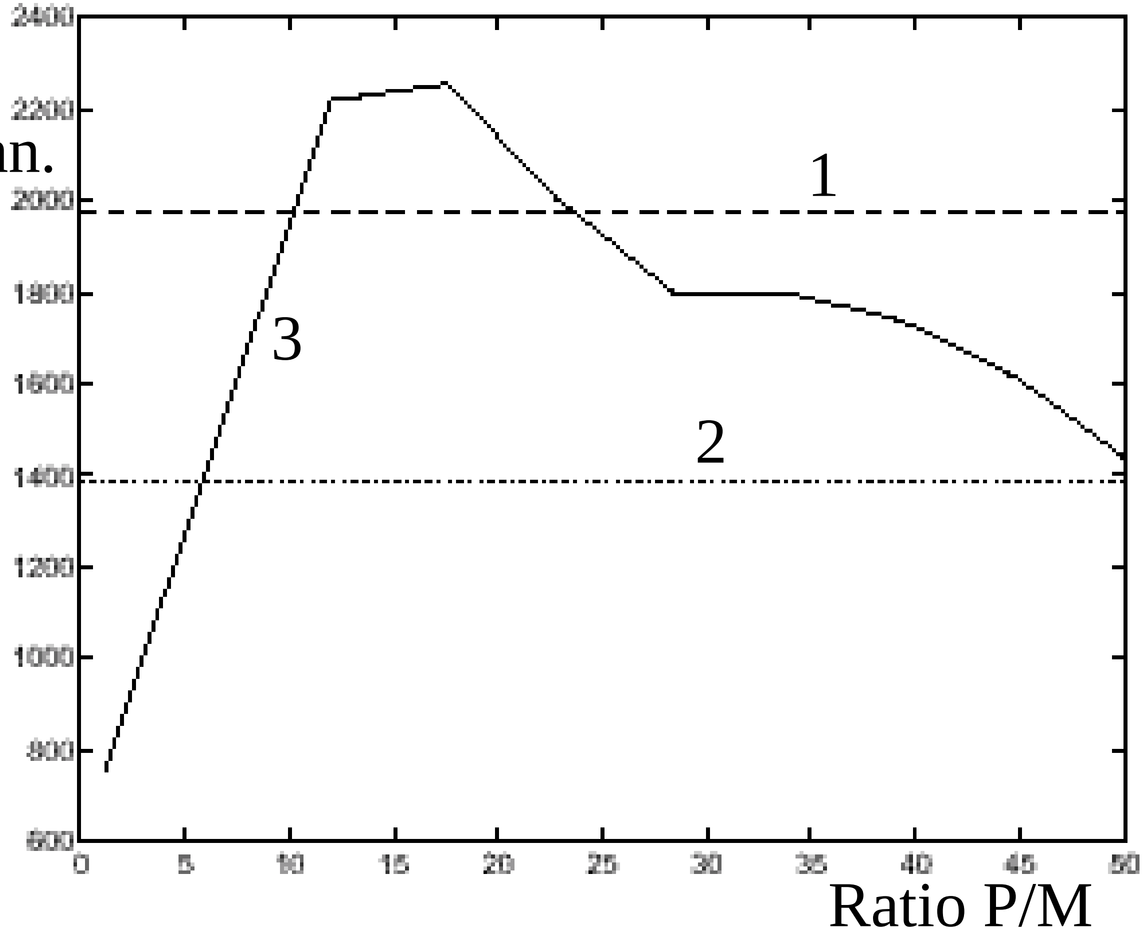
B=0.05 MB,X=1Mb

Input:

p=0.2Mbps

b=0.05

Mean rate m=0.05

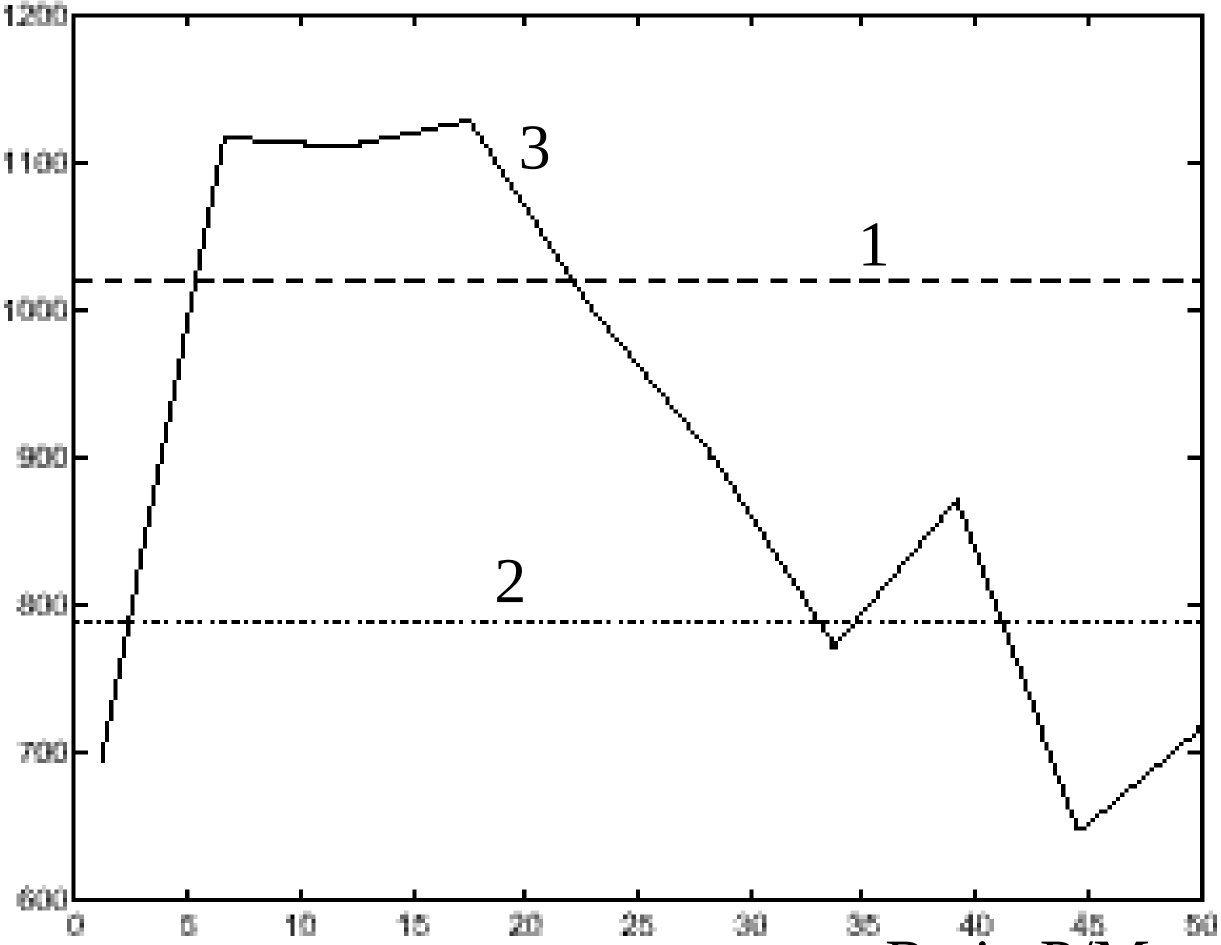


Performance of VBR over VBR traffic aggregation can significantly exceed the performance of CBR aggregation

VBR Multiplexor

N. conn.

Mean rate m=0. 1



Ratio P/M

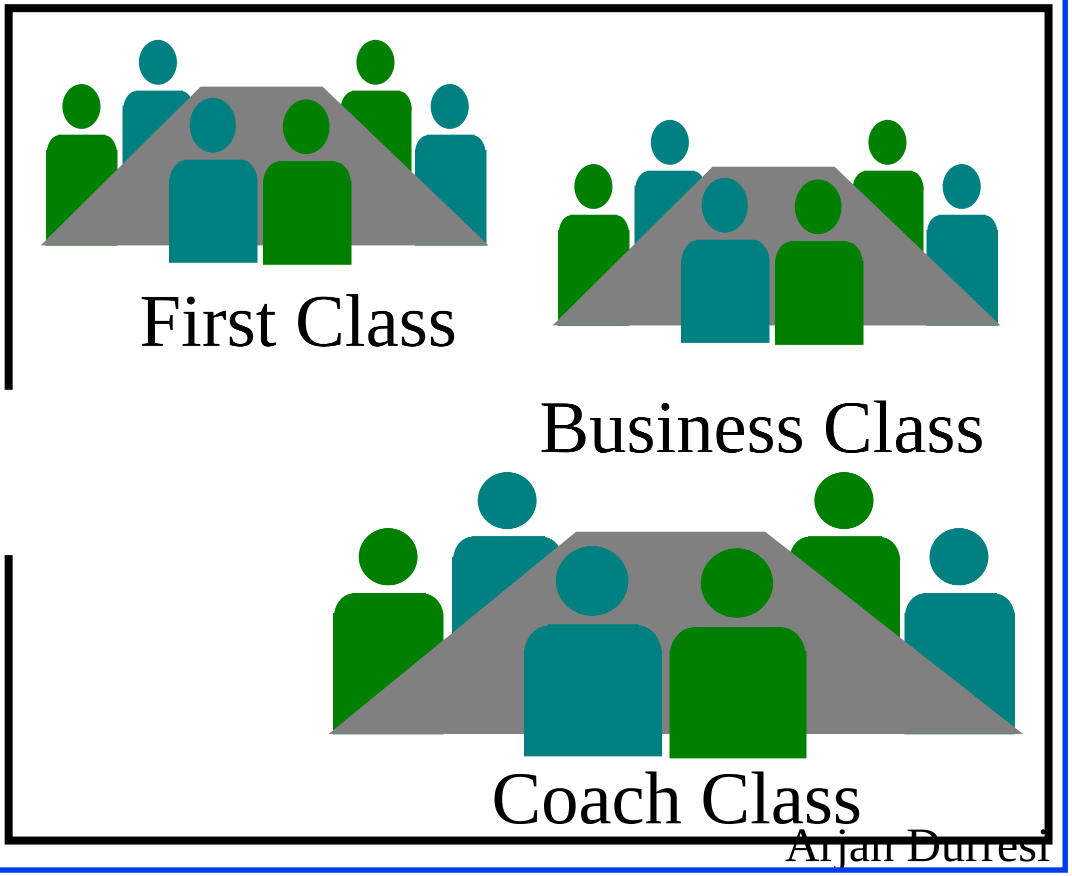
Task 5: Summary

- Why IP multiplexing?
- Models for input and output flows and multiplexing rules: find an optimized solution
- Leaky Bucket Model
- D-BIND/H-BIND Model
- Deterministic Approaches
- Stocastic/Statistical Approaches
- VBR over VBR better than CBR over CBR
- Simulations using Leaky Bucket Model

Task 6: Study of Assured Forwarding in Satellite Networks

- Key Variables
- Buffer Management Classification: Types of RED
- Traffic Types and Treatment
- Level of Reserved Traffic
- Two vs Three: Best Results
- Summary

Key Variables

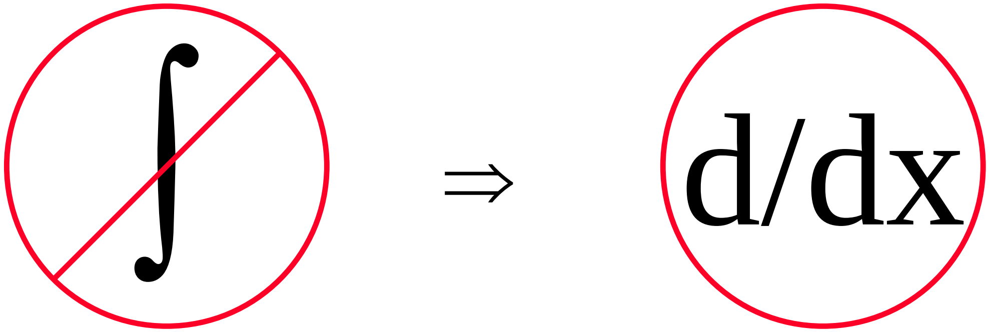


First Class

Business Class

Coach Class

Differentiated Services



- DiffServ to standardize IPv4 ToS byte’s first six bits
- Packets gets marked at network ingress
  Marking ⇒ treatment (behavior) in rest of the net
  Six bits ⇒ 64 different per-hop behaviors (PHB)

Ver	Hdr Len	Type of Service (ToS)	Tot Len
4b	4b	8b	16b

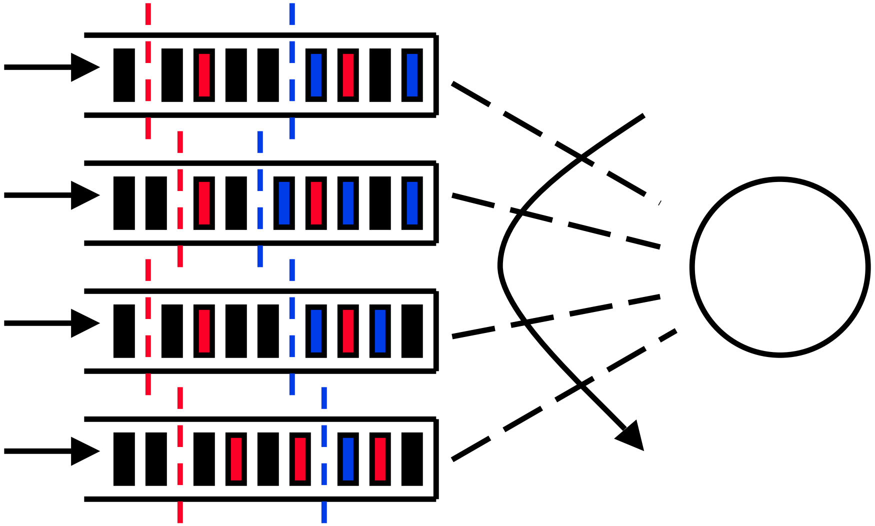
DiffServ (Cont)

- Per-hop behavior = % of link bandwidth, Priority
- Services: End-to-end. Voice, Video, ...
  - Transport: Delivery, Express Delivery,...
    Best effort, controlled load, guaranteed service
- DS group will not develop services
  They will standardize “Per-Hop Behaviors”
- Marking based on static “Service Level Agreements” (SLAs). Avoid signaling.

Expedited Forwarding

- Also known as “Premium Service”
- Virtual leased line
- Similar to CBR
- Guaranteed minimum service rate
- Policed: Arrival rate < Minimum Service Rate
- Not affected by other data PHBs
  ⇒ Highest data priority (if priority queueing)
- Code point: 101 110

Assured Forwarding



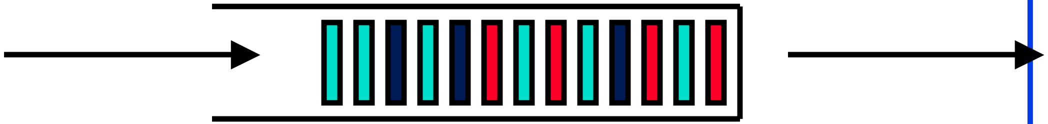
- PHB Group
- Four Classes: No particular ordering
- Similar to nrt-VBR/ABR/GFR

Key Variables

- **Bandwidth Management**:
  - Number of colors: One, Two, or Three
  - Percentage of green (reserved) traffic: Low, high, oversubscribed
- **Buffer Management**:
  - Tail drop or RED
  - RED parameters, implementations
- **Traffic Types and their treatment**:
  - Congestion Sensitivity: TCP vs UDP
  - Excess TCP vs Excess UDP
- **Network Configuration**:

  Our goal is to identify results that apply to all configs.

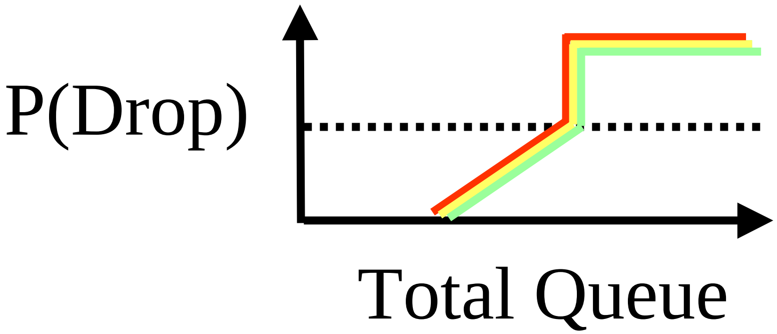
Buffer Management Classification



- Accounting (queued packets):
  Per-color, per-VC, per-flow, or Global
  Multiple or Single
- Threshold: Single or Multiple
- Four Types:
  - Single Accounting, Single threshold (SAST)
  - Single Accounting, Multiple threshold (SAMT)
  - Multiple Accounting, Single threshold (MAST)
  - Multiple Accounting, Multiple threshold (MAMT)

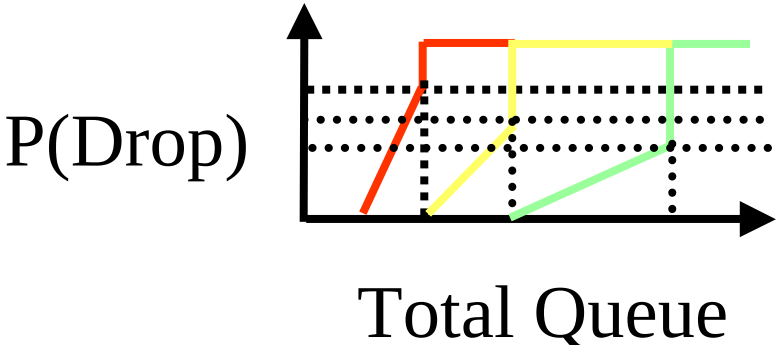
Types of RED

- Single Accounting Single Threshold (SAST): Color-blind Random Early Discard (RED)



Used in present diffserv-unaware routers

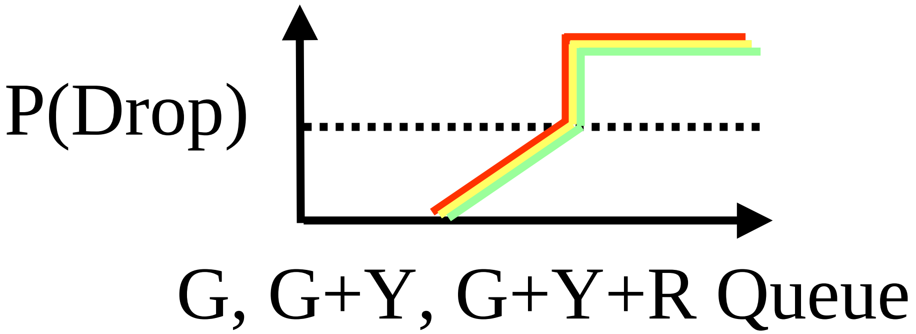
- Single Accounting Multiple Threshold (SAMT): Color-Aware RED as implemented in some products



Used in this study

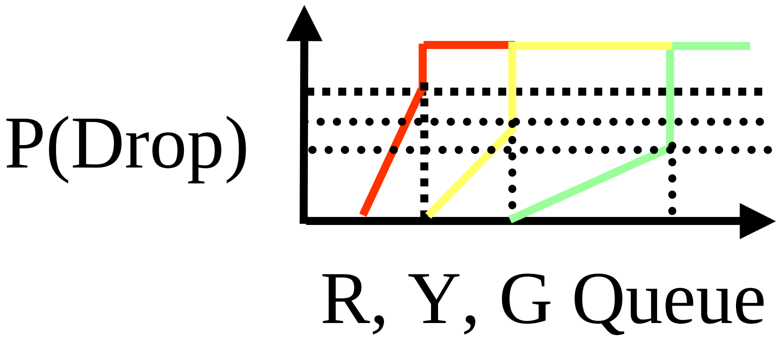
Types of RED (Cont)

- Multiple Accounting Single Threshold (MAST):



Used in our previous study

- Multiple Accounting Multiple Threshold (MAMT):



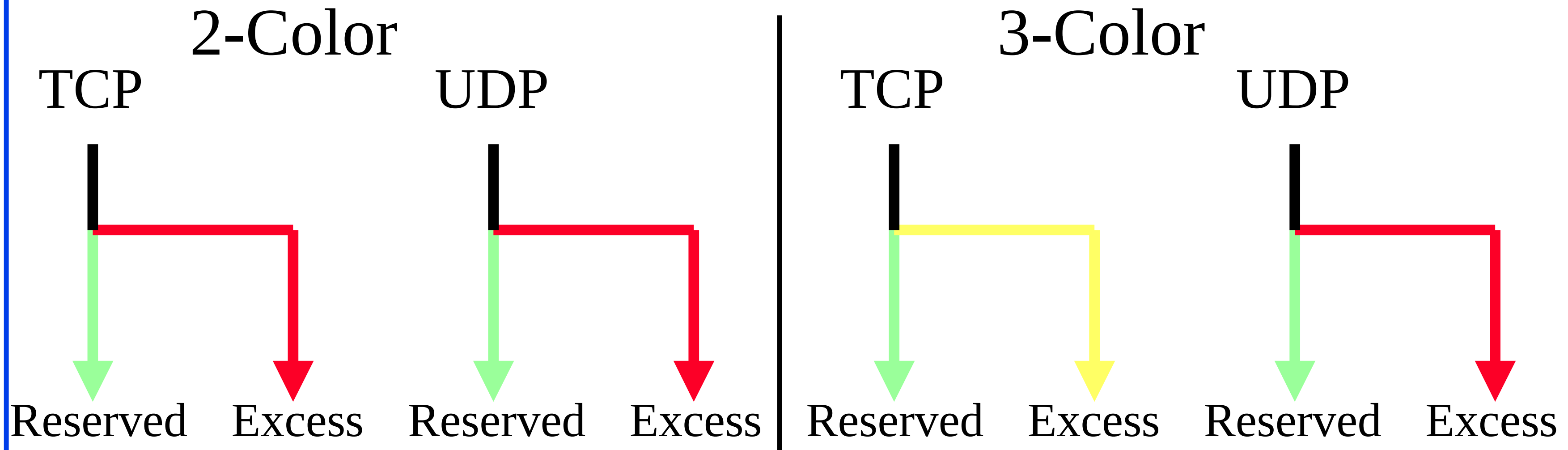
Conclusion:
More Complexity
⇒ More Fairness

Traffic Types and Treatment

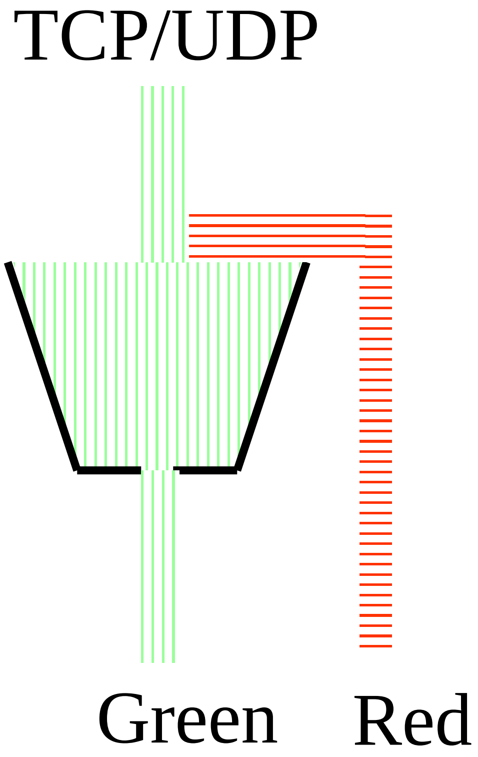
- Both TCP and UDP get their reserved (green) rates
- Excess TCP competes with excess UDP
- UDP is aggressive

  ⇒ UDP takes over all the excess bandwidth

  ⇒ Give excess TCP better treatment than excess UDP



Two Drop Precedences



TCP/UDP

Committed
Burst
Size (CBS)

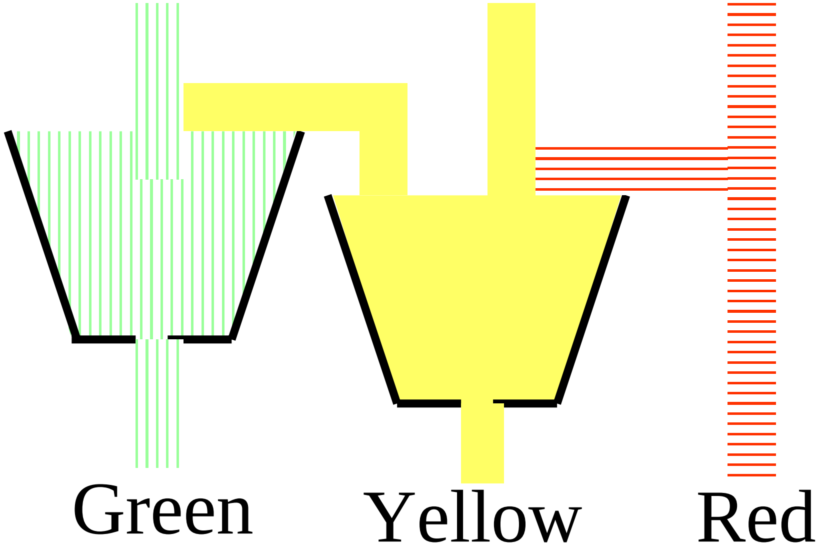
Committed
Information
Rate (CIR)

Green

Red

- All packets up to CIR are marked Green
- Overflowed packets are marked Red

Three Drop Precedences

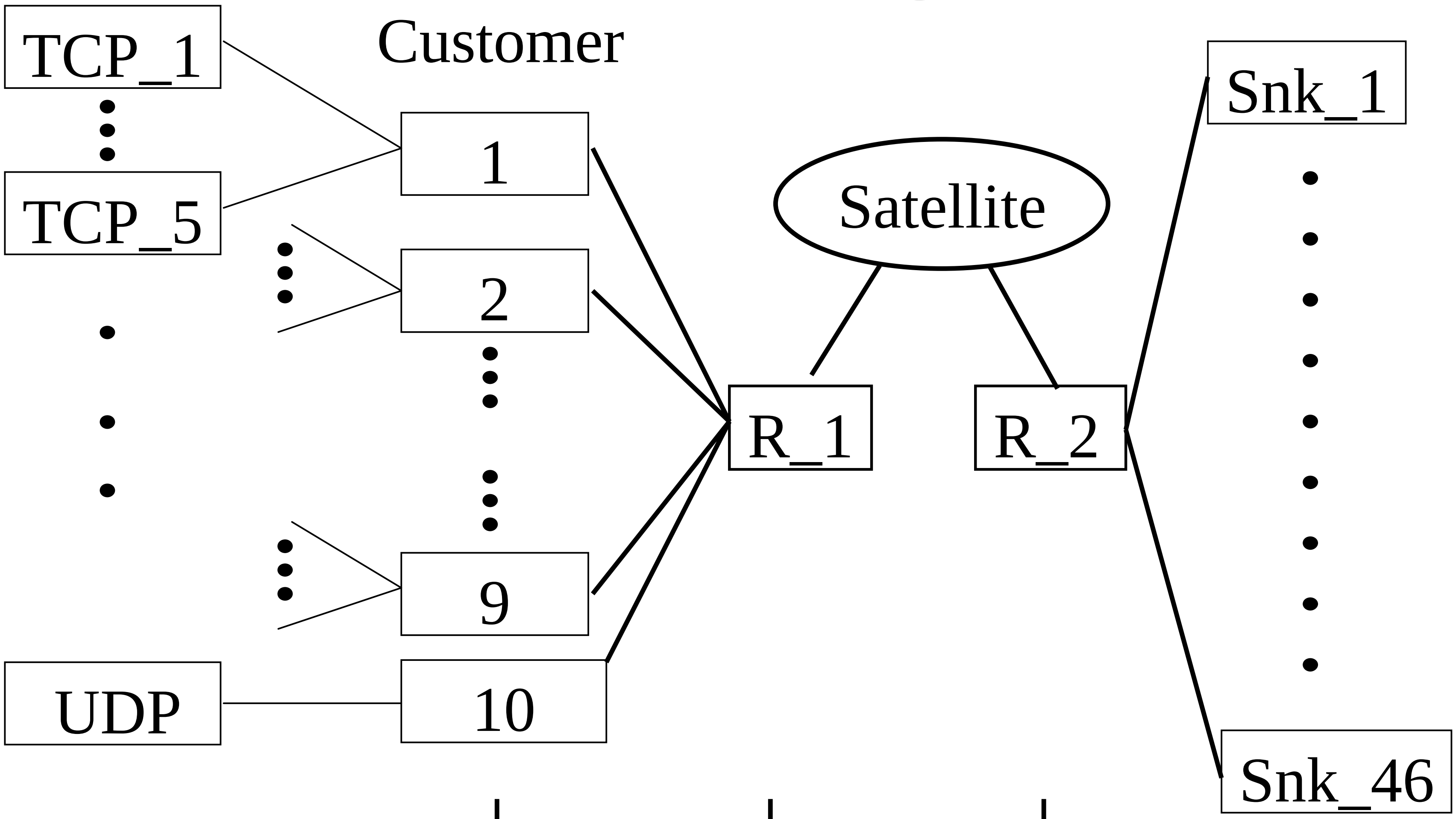


- Tokens in Green, Yellow buckets are generated independently.
- Parameters: Token generation rate and Bucket Size for Green and Yellow buckets
- Color Aware ⇒ Excess packets overflow to next color

Level of Reserved Traffic

- Percentage of reserved (green) traffic is the most important parameter
- If the green traffic is high
  ⇒ No or little excess capacity
  ⇒ Two or three colors perform similarly
- If the green traffic is low
  ⇒ Lots of excess capacity
  ⇒ Behavior of TCP vs UDP impacts who gets excess
  ⇒ Need 3 colors + Need to give excess TCP yellow + Need to give excess UDP red colors

Simulation Configuration



TCP_1

TCP_5

UDP

Customer

1

2

9

10

R_1

R_2

Satellite

Snk_1

Snk_46

1 µs	5 µs	250 ms	5 µs
10 Mbps	1.5 Mbps	1.5 Mbps	1.5 Mbps

Link Parameters

	Link B/W	Link Delay	Link Policy
Between TCP/UDP & Customer i	10 Mbps	1 µs	DropTail
From Customer i to R1	1.5 Mbps	5 µs	DropTail w marker
From R1 to Customer i	1.5 Mbps	5 µs	DropTail
From R1 to R2	1.5 Mbps	250 µs	RED_n
From R2 to R1	1.5 Mbps	250 µs	DropTail
Between R2 and sink i	1.5 Mbps	5 µs	DropTail

Simulation Parameters

- S*ingle* A*ccounting* M*ultiple* T*hreshold* RED
- RED Queue Weight for All Colors: w = 0.002

$$Q_{avg} = (1\text{-}w)Q_{avg} + w\ Q$$

- Maximum Queue Length (For All Queues): 60 packets
- TCP flavor: Reno
- TCP Maximum Window: 64 packets
- TCP Packet Size: 576 bytes
- UDP Packet Size: 576 bytes
- UDP Data Rate: 1.28Mbps

Two Color Simulations

Simulation Configuration	Green Token Generation Rate [kbps]	Green Token Bucket Size (in Packets)	Maximum Drop Probability {Green, Red}	Drop Thresholds {Green, Red}

1 Through 1152	12.8,
	25.6,
	38.4,
	76.8,
	102.4,
	128,
	153.6,
	179.2
		1,
		2,
		4,
		8,
		16,
		32
			{0.1,0.1}
			{0.1,0.5}
			{0.1,1}
			{0.5,0.5}
			{0.5,1}
			{1,1}
				{40/60,0/10}
				{40/60,0/20}
				{40/60,0/5}
				{40/60,20/40}

Three Color Simulations

Simulation Config.	Green Token Gener. Rate [kbps]	Green Token Bucket Size in Packets	Yellow Token Bucket Size in Packets	Max Drop Probability {Green, Yellow, Red}	Drop Thresholds {Green, Yellow, Red}	Yellow Token Gener. Rate [kbps]

1 Through 2880	12.8,
	25.6,
	38.4,
	76.8
		1,
		2,
		4,
		8,
		16,
		32
			1,
			2,
			4,
			8,
			16,
			32
				{0.1,0.5,1}
				{0.1,1,1}
				{0.5,0.5,1}
				{0.5,1,1}
				{1,1,1}
					{40/60,20/40, 0/10}
					{40/60,20/40, 0/20}
						128,
						12.8

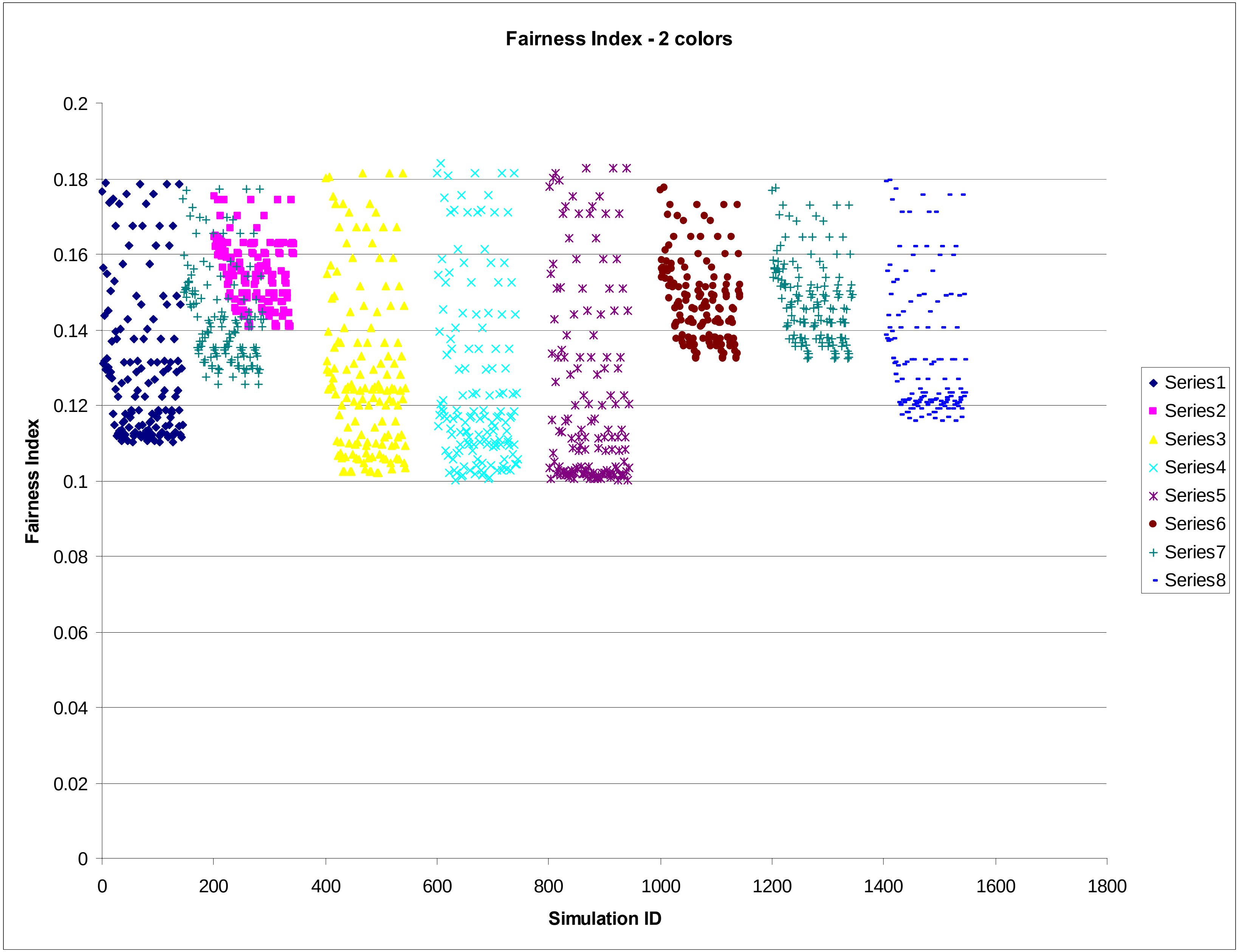
Fairness Index

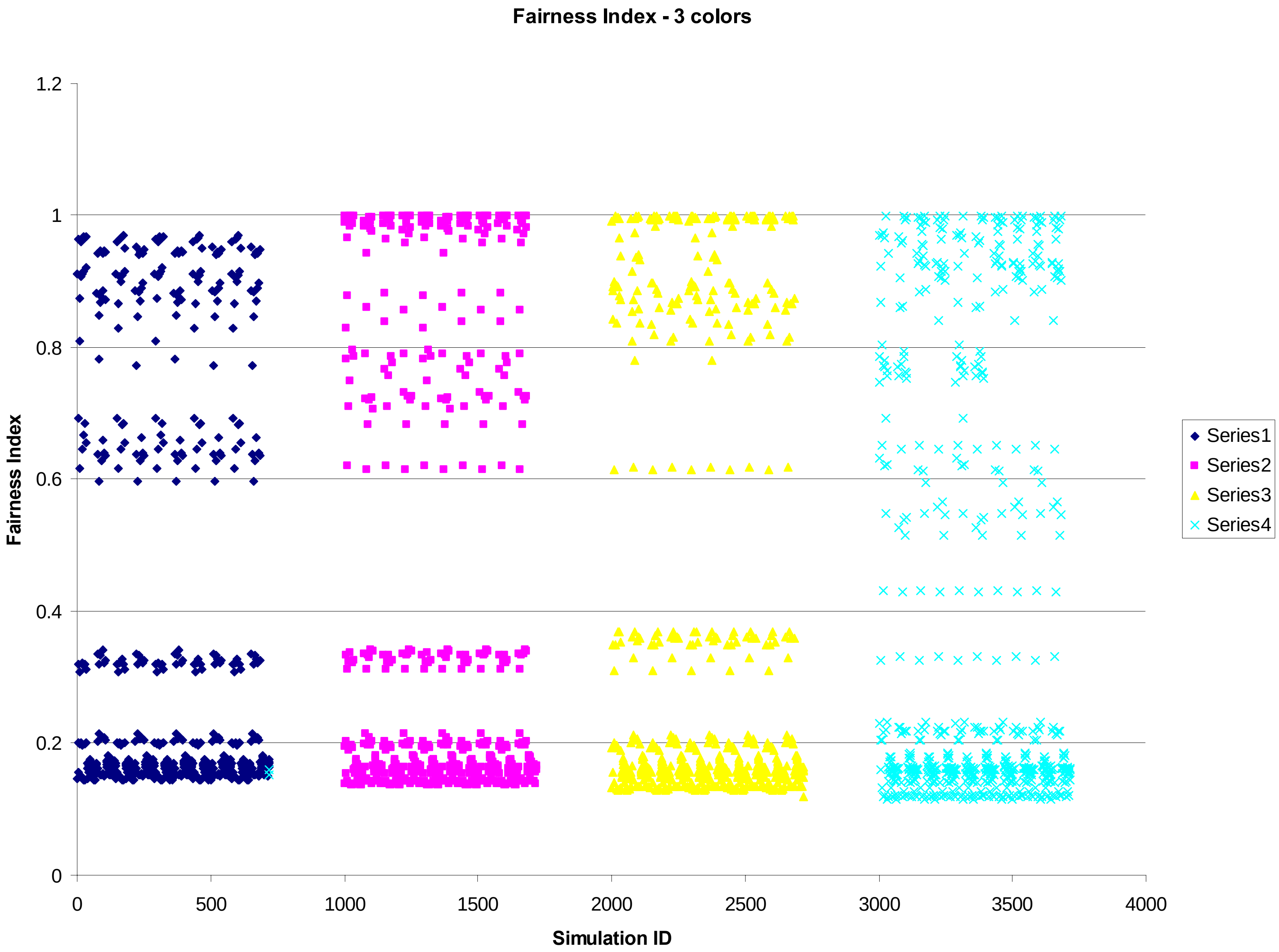
- Measured Throughput: $(T_1, T_2, \ldots, T_n)$
- Use any criterion (e.g., max-min optimality) to find the Fair Throughput $(O_1, O_2, \ldots, O_n)$
- Normalized Throughput: $x_i = T_i/O_i$

$$\text{Fairness Index} = \frac{(\Sigma\, x_i)^2}{n\Sigma\, x_i^2}$$

Example: 50/50, 30/10, 50/10 ⇒ 1, 3, 5

$$\text{Fairness Index} = \frac{(1+3+5)^2}{3(1^2+3^2+5^2)} = \frac{9^2}{3(1+9+25)} = 0.81$$


Fairness Index - 2 colors
Fairness Index
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0
200
400
600
800
1000
1200
1400
1600
1800
Simulation ID
Series1
Series2
Series3
Series4
Series5
Series6
Series7
Series8


Fairness Index - 3 colors
1.2
1
0.8
0.6
0.4
0.2
0
Fairness Index
0
500
1000
1500
2000
2500
3000
3500
4000
Simulation ID
◆ Series1
■ Series2
▲ Series3
× Series4

ANOVA

- Analysis of Variance (ANOVA) - Statistical tool
- Most Important Factors Affecting Fairness and Throughput:
  - What % of the Variation is explained by Green (Yellow) rate?
  - What % of the Variation is explained by Bucket Size ?
  - What % of the Variation is explained the Interaction between Green (Yellow) Rate and Bucket Size

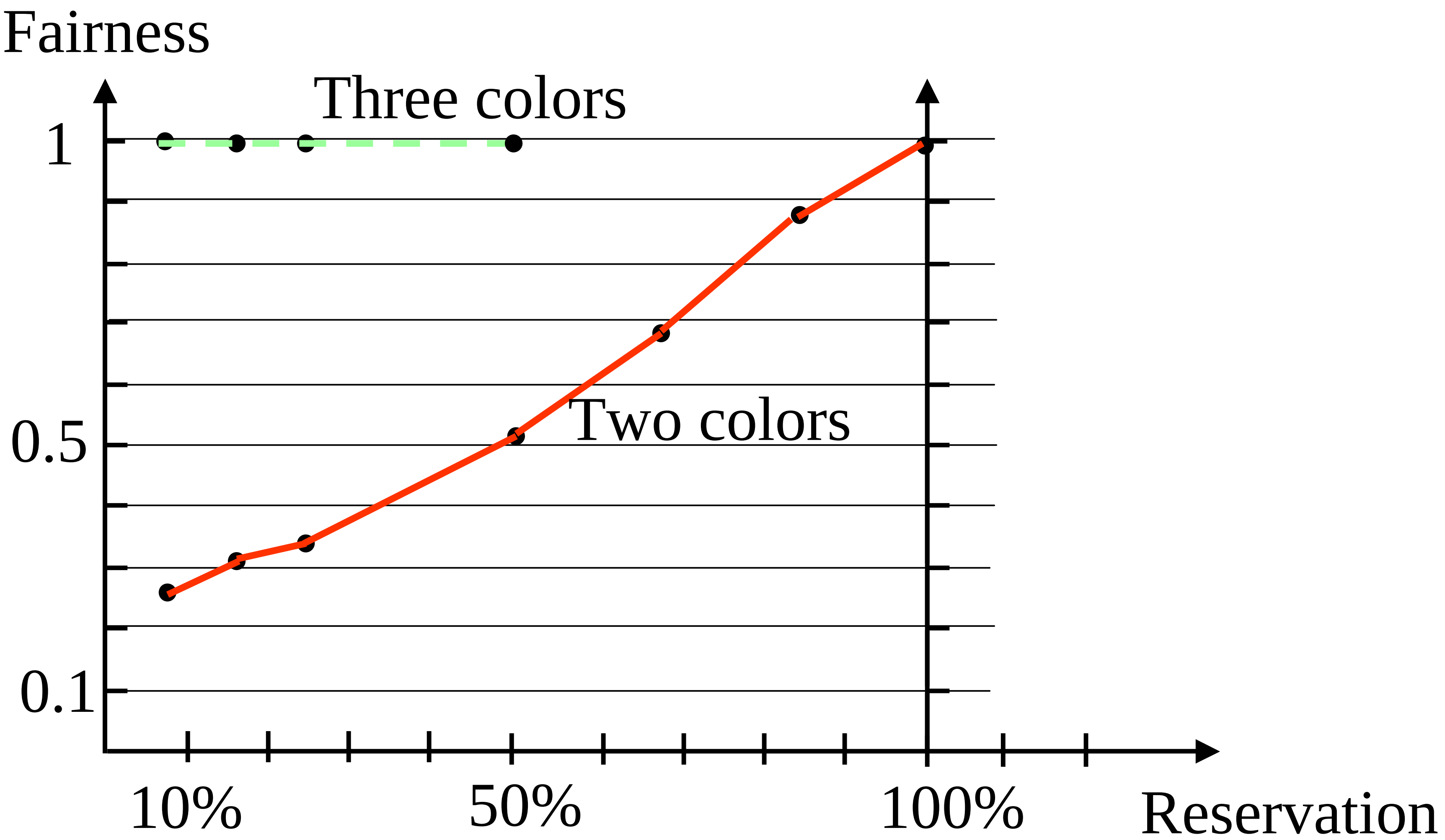
ANOVA For 2 Color Simulations

- Most Important Factors Affecting Fairness:
    - Green Rate (Explains 65.6% of the Variation)
    - Bucket Size (Explains 19.2% of the Variation)
    - Interaction between Green Rate and Bucket Size (Explains 14.8% of the Variation)

ANOVA For 3 Color Simulations

- Most Important Factors Affecting Fairness:
  - Yellow Rate (Explains 41.36% of the Variation)
  - Yellow Bucket Size (Explains 28.96% of the Variation)
  - Interaction Between Yellow Rate And Yellow Bucket Size (Explains 26.49% of the Variation)

Two vs Three: Best Results



Task 6: Summary

1. The key performance parameter is the level of green (reserved) traffic
2. If reserved traffic level is high or if there is any overbooking, two and three colors give the same throughput and fairness
3. If the reserved traffic is low, three colors give better fairness than two colors
4. Classifiers have to distinguish TCP and UDP: Reserved TCP/UDP ⇒ Green, Excess TCP ⇒ Yellow, Excess UDP ⇒ Red
5. RED parameters and implementations have significant impact.